



Sistema de informação corporativo recorrendo a Realidade Aumentada

JOÃO FILIPE OLIVEIRA VAZ

Outubro de 2020

Sistema de informação corporativo recorrendo a Realidade Aumentada

João Filipe Oliveira Vaz

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Sistemas Gráficos e Multimédia**

Orientador: António Vieira de Castro (PhD)

Porto, outubro 2020

À minha família por me ter apoiado em todos os momentos ao longo da vida

Resumo

A evolução das tecnologias de Realidade Aumentada (RA) em muito tem vindo a contribuir em diversas áreas como o entretenimento, o turismo e o ensino.

Processos que antes teriam de ser feitos de forma manual ou com contacto direto, como por exemplo guiar utilizador para um determinado local ou apresentar informação, pode agora ser feita automaticamente, acrescentando novas informações e objetos virtuais ao mundo real.

Este documento contém toda a informação referente ao conteúdo elaborado durante a unidade curricular Tese / Dissertação / Estágio, do Mestrado em Engenharia Informática, do Departamento de Engenharia Informática (DEI) do Instituto Superior de Engenharia do Porto (ISEP).

O projeto em questão consistiu em criar uma prova de conceito para investigar o potencial da RA como meio de orientação dentro de espaços pequenos e/ou fechados e se esta é fácil e intuitiva.

Para o efeito foi desenvolvido um protótipo de uma aplicação móvel que proporciona aos utilizadores a visualização do mapa do DEI. A partir daí, os utilizadores podem escolher um destino dentro do ISEP e ser guiados até ele através de setas de orientação virtuais.

Dado o facto de estarmos presentemente numa fase de pandemia, o protótipo foi alvo de vários testes, mas numa perspetiva funcional, pelo seu autor.

O presente documento termina com as conclusões e o trabalho futuro.

Palavras-chave: Realidade Aumentada, Geolocalização, Android, Mobile, Localização
nterior

Abstract

The evolution of Augmented Reality (AR) technology has greatly contributed to many areas like entertainment, tourism, and education.

Processes that needed to be done in a manual way or with physical presence, for example guiding a user through a determined location or presenting information, can now be automated by adding new information and virtual objects to the real world.

This document contains all the information related to the content elaborated during the curricular unit Tese / Dissertação / Estágio, of the master's degree in Informatic Engineering of the Informatic Engineering Department (DEI) of Instituto Superior de Engenharia do Porto (ISEP).

The project consisted in creating a proof of concept to investigate the AR potential as a method of orientation inside of small and/or closed spaces and if it's easy and intuitive. For that effect it was developed a prototype of a mobile application that provides the users a visualization of the DEI ISEP's map. From there, the users can choose a destination inside of ISEP and be guided to it through virtual orientation arrows.

Given the fact that we are currently in a pandemic phase, the prototype has undergone several tests, but in a functional perspective, by its author.

This document ends with the conclusions and future work.

Keywords: Augmented Reality, Geolocation, Android, Mobile, Indoor Location

Agradecimentos

Quero utilizar esta secção para agradecer a todos aqueles que de alguma forma enriqueceram o meu percurso académico.

Agradeço em primeiro lugar à instituição (ISEP) e ao departamento (DEI) por todo o conhecimento que lá adquiri no período tanto de licenciatura como de Mestrado.

Agradeço também ao meu orientador, o Prof. Doutor António Vieira de Castro, docente do DEI, investigador no GILT (Games Interaction and Learning Technologies), diretor do LAMU (Laboratório Multimédia), Membro do EIPP (Unidade de Inovação Pedagógica do Politécnico do Porto) e Membro do CCRAV (Centro de Competências em Realidade Aumentada e Virtual) do Politécnico do Porto por toda a sua dedicação, apoio e incentivo ao longo do presente estudo.

Ao LAMU (Laboratório Multimédia do Departamento de Engenharia Informática do ISEP) pelo apoio ao desenvolvimento do presente estudo.

Ao João Coelho por me ter acompanhado nesta fase final e a todos os meus colegas e amigos por todo o apoio e companhia prestados em todos os momentos desta etapa, por todos os momentos bons e menos bons que passei sempre com a vossa companhia. A todos os professores que fizeram parte deste meu percurso académico, por tudo o que me transmitiram ao longo destes cinco anos de formação.

Ao NEI (Núcleo de Estudantes de Informática) do ISEP, do qual tive o privilégio de ser presidente por todas as aprendizagens paralelas e conhecimentos que me proporcionou.

Por fim agradeço à minha família que me acompanhou, educou e cuidou de mim principalmente durante estes últimos cinco anos.

Índice

1	Introdução.....	1
1.1	Enquadramento	1
1.2	Problema.....	2
1.3	Objetivos e contributos esperados.....	2
1.4	Motivação	3
1.5	Estrutura do documento.....	4
2	Contexto e Estado da Arte	7
2.1	A Realidade Aumentada	7
2.1.1	Realidade Aumentada Ligada à Geo Localização	11
2.1.2	Plickers (RA em Educação)	12
2.1.3	Story of the Forest.....	13
2.1.4	Skin and Bones.....	13
2.1.5	O Google Maps	14
2.1.6	O Pokemon Go.....	16
2.1.7	Conclusões sobre o estado da arte	17
2.2	Contextualização Tecnológica	18
2.2.1	O Android Studio	18
2.2.2	Motores de jogo	19
2.2.3	SDK de Realidade Aumentada	21
2.2.4	Tecnologias de orientação no espaço	25
2.2.5	Conclusões sobre as tecnologias	27
2.3	Análise de valor.....	27
2.3.1	Processo de Inovação	29
2.3.2	Quadro de Modelo de Negócio	33
2.3.3	QFD	38

3	Implementação	43
3.1	Contexto da solução	43
3.2	Análise e Design da solução	45
3.2.1	Os Casos de Uso	46
3.2.2	Diagramas de Arquitetura.....	47
3.3	A Orientação no espaço	49
3.3.1	A Utilização do Indoor Atlas.....	50
3.4	Aplicação Mobile	54
3.4.1	A Utilização do Android Studio	55
3.5	A Implementação da Realidade Aumentada.....	60
3.6	Adaptações.....	66
3.7	Resumo.....	66
4	A Avaliação da Solução	69
4.1	QEF	69
4.2	A Metodologia de Avaliação.....	70
5	Conclusões e trabalho futuro	75
5.1	Conclusões.....	75
5.2	Trabalho futuro	76
6	Referências.....	77

Lista de Figuras

Figura 1 – Sensorama.....	8
Figura 2 Primeiro protótipo de HDM, “The Ultimate Display”	9
Figura 3 Google Glasses	10
Figura 4 – O Hololens da MicroSoft	11
Figura 5 - Utilização do Plickers	12
Figura 6 - Skin and Bones.....	14
Figura 7 - Google Maps.....	15
Figura 8 – Vista do ISEP pelo Google Maps	16
Figura 9 – Vista do ISEP através do modo de Street View.....	16
Figura 10 – Mapa do pokemon GO.....	17
Figura 11 - Logotipo do unreal Engine	19
Figura 12 – Logotipo do Unity.....	21
Figura 13 - Fórmula do valor.....	27
Figura 14 - Fórmula do valor de mercado	28
Figura 15 – Processo de inovação do produto	30
Figura 16 – New Concept Development Model	31
Figura 17 – Estrutura do <i>Business Model Canvas</i>	34
Figura 18 - House of Quality	41
Figura 19 – Objetivo da aplicação.....	44
Figura 20 – Orientação do utilizador ao longo do percurso	44
Figura 21 – Esquema que depreende que o percurso realizado acontece dentro do ISEP	45
Figura 22 - Diagrama de casos de uso	46
Figura 23 - Primeiro diagrama de arquitetura da solução.....	48
Figura 24 - Segundo diagrama de arquitetura da solução	48
Figura 25 - Pagina principal do Indoor Atlas.....	51
Figura 26 – Interface para alinhamento dos planos dos pisos no Indoor Atlas	52
Figura 27 – Mapa com os Waypoints criados	52

Figura 28 – Cobertura magnética do DEI	53
Figura 29 – Plugin do Indoor Atlas para Unity	55
Figura 30 - Configuração de gradle necessária para utilizar a API Indoor Atlas	56
Figura 31 – Código para carregar o bitmap.....	57
Figura 32– Código para inserir o bitmap no mapa.....	58
Figura 33– Código que trata da mudança de localização	58
Figura 34– Código para mudar marcadores.....	59
Figura 35 - Configuração de gradle necessária para utilizar o ArCore	60
Figura 36 – Implementação do OnDrawFrame	61
Figura 37 – Código para criar as setas de orientação	62
Figura 38 – Verificação da localização do utilizador	63
Figura 39 – Apresentação da rota no mapa	64
Figura 40 – Iniciação da deslocação	65
Figura 41– Conclusão da deslocação.....	65
Figura 42 - Qualitative Evaluation Framework.....	70

Lista de Tabelas

Tabela 1 – Comparação entre SDK de AR	24
Tabela 2- Benefícios e sacrifícios	29
Tabela 3 – Quadro de Modelo de Negócio	34
Tabela 4 - Métricas de avaliação da Funcionalidade	71
Tabela 5 - Métricas de avaliação da interação com o utilizador	71
Tabela 6 - Métricas de avaliação da qualidade do conteúdo	72
Tabela 7 - Métricas de avaliação da Adaptabilidade	72
Tabela 8 – Métricas de avaliação da Eficiência	73

Acrónimos

API	Interface de programação de aplicações
AR	Augmented Reality/Realidade Aumentada
DEI	Departamento de Engenharia Informática
GPS	Sistema de posicionamento global
FFE	Fuzzy Front End
IDE	Ambiente de desenvolvimento integrado
ISEP	Instituto Superior de Engenharia Informática
JPEG	Joint Photographic Experts Group
MVP	Minimum value product
NCD	New Concept Development
NPD	New Product Development
PNG	Portable Network Graphics
QEF	Qualitative Evaluation Framework
QFD	Quality Function Deployment
SDK	Kit de desenvolvimento de software

Glossário

Anchor	Descreve uma orientação e localização física no mundo real
Brainstorm	atividade desenvolvida para explorar a potencialidade criativa de um indivíduo ou de um grupo
Customer Chain	Cadeia de indivíduos que irão comprar um produto
Geolocalização	Identificação ou estimativa da localização geográfica do mundo real de um objeto
Marker	Pista visual que desencadeia a visualização de informação virtual
Open-source	Software de Código aberto ao público geral
Plugin	Programa de computador usado para adicionar funções a outros programas maiores
Quaternion	Extensão do conjunto dos números complexos
Widget	Elemento de interação numa interface gráfica
WayPoint	Ponto no globo terrestre precisamente definido por coordenadas geográficas e que é facilmente reconhecível
WayFinding	Formas de uma pessoa se orientar num espaço físico e navegar de um espaço para outro

1 Introdução

“O sábio nunca diz tudo o que pensa, mas pensa sempre tudo o que diz”

Aristóteles

Neste primeiro capítulo é feito o enquadramento do tema da tese, evidenciando a ligação entre a orientação no espaço e o aparecimento de novas oportunidades devido ao aparecimento de novas tecnologias, como a Realidade Aumentada.

De seguida são identificadas as principais dificuldades e problemas existentes, sentidos pelos novos alunos do DEI quando se encontram pela primeira vez neste.

Tendo em conta a resolução destes problemas, são depois definidos os objetivos que visam o desenvolvimento de um protótipo com potencial para aplicar técnicas de Realidade Aumentada associadas ao processo de geolocalização para proporcionar ao utilizador a rota para chegar ao seu destino.

Por último é apresentada a motivação para abraçar o presente estudo, assim como a organização do presente documento.

1.1 Enquadramento

Desde os tempos primórdios um dos grandes desafios do ser humano passa por descobrir e documentar a sua localização assim como orientar-se pelo espaço disponível, sendo até os dias de hoje uma área que está constantemente a ser explorada e melhorada.

Nesta dissertação ir-se-á abordar a Realidade Aumentada e o seu uso para a otimização na ajuda da orientação no espaço.

A Realidade Aumentada (RA) é uma tecnologia recente que consiste em colocar elementos virtuais num ambiente real e dá a possibilidade ao utilizador de interagir com esses elementos da mesma maneira que interagiria com a sua contraparte real.

A base de trabalho para a realização desta dissertação será o edifício B do Instituto Superior de Engenharia do Porto(ISEP), onde se encontra o Departamento de Engenharia Informática(DEI).

1.2 Problema

Sempre que um novo estudante entra no DEI pela primeira vez, sem ter qualquer conhecimento prévio sobre o edifício, tende a sentir-se perdido necessitando, por isso, de alguma forma de se orientar neste novo espaço. Com o aparecimento da Realidade Aumentada torna-se possível a integração de elementos virtuais num ambiente real através do auxílio de, por exemplo, um telemóvel com camara fotográfica e giroscópio. Desta maneira é pretendido realizar-se um estudo sobre a possibilidade da utilização de Realidade Aumentada para associar informações em tempo real sobre as instalações e equipamentos que estejam dentro do departamento e desta forma tornar mais interessante e otimizada a obtenção de informação sobre o espaço que o rodeia.

De acordo com o problema descrito, detêm-se que a hipótese que esta tese pretende apresentar soluções através de uma prova de conceito é:

Será possível obter uma fácil orientação e perceção do ambiente envolvente em espaços fechados e de pequenas dimensões utilizando Realidade Aumentada?

1.3 Objetivos e contributos esperados

Considerando o potencial da Realidade Aumentada para acrescentar elementos virtuais ao ambiente real, pretende-se construir uma ferramenta que permita guiar o utilizador pelo ambiente onde se encontra e fornecer-lhe mais informações sobre o que o rodeia. Para o efeito será essencial efetuar uma análise do estado atual de soluções desta natureza, assim como identificar as diversas tecnologias e procedimentos com potencial para serem utilizadas de forma a guiar o utilizador num determinado local. Pretende-se também desta maneira perceber o fenómeno da Realidade Aumentada adaptado à geolocalização e informação.

Após uma análise prévia de tecnologias e ferramentas com potencial para desenvolver uma solução desta natureza, serão sistematizadas e avaliadas soluções de negócio existentes, assim como o potencial da nova solução e a sua avaliação no sentido de aferir se é adequada à resolução do problema e a avaliação do potencial do modelo proposto. Pretende-se desenvolver uma solução, baseada no Departamento de Engenharia Informática do ISEP, que permita ao utilizador perceber onde se encontra e como chegar ao seu destino, assim como descobrir mais informações sobre o local que o rodeia. Para o efeito, será necessário:

- Analisar soluções existentes;
- Entender e estudar as ferramentas utilizadas para o desenvolvimento AR;
- Identificar bibliotecas, API's e linguagens com potencial para o presente estudo;
- Explorar e apreender os conceitos e técnicas geolocalização;
- Explorar e apreender os conceitos de Marker Based AR;
- Desenvolver e implementar a solução;
- Implementação de mecanismos de interação do utilizador com o mundo virtual;
- Testar a aplicação;
- Validar aspetos técnicos e funcionais para ajudar a resolver o problema.

1.4 Motivação

Sendo o mestrando licenciado em Engenharia Informática, desde sempre lidou de perto com as novas tecnologias, tendo acompanhado ainda mais a sua evolução nos últimos anos e relevando elevado interesse pelo uso de tecnologias emergentes, como é o caso dos dispositivos móveis e da Realidade Aumentada.

Tendo já feito um projeto/estágio na área da Realidade Aumentada e virtual, é para ele uma forma de aprofundar ainda mais o seu conhecimento nestas tecnologias, que lhe serão uma mais valia tanto a nível pessoal como a nível profissional.

Paralelamente sempre demonstrou interesse pela orientação no espaço, quer em espaços abertos como fechados, assim como o uso da geolocalização, quer para a orientação, quer para outros fins como por exemplo o entretenimento. Assim sendo é um

privilégio para o mestrando abraçar um projeto desta natureza, onde é possível conjugar o uso da Realidade Aumentada com técnicas de geolocalização e orientação em espaços fechados.

Por outro lado é também bastante motivante poder realizar este estudo numa instituição como o ISEP que é um instituto de renome mundial e que tem fama de ir mais longe que o resto e de ser dos primeiros a abraçar novas inovações e oportunidades. Sendo também um local em que todos os anos entram novos alunos e são feitas várias convenções, será uma mais valia apostar numa solução que ajude a orientar estes indivíduos e desta maneira tornar muito mais agradável a sua visita ao ISEP.

Partindo do pressuposto que o protótipo terá sucesso, poder pensar em alargar o projeto para o resto das instituições de ensino do Porto, assim como outros edifícios como por exemplo a camara do Porto, é também um fator que faz com que o mestrando se esforce ainda mais para conclusão do projeto.

1.5 Estrutura do documento

Este documento encontra-se dividido em 5 capítulos.

No primeiro capítulo é feito o enquadramento do tema da tese, evidenciando a ligação entre a orientação no espaço e o aparecimento de novas oportunidades devido ao aparecimento de novas tecnologias, como a Realidade Aumentada. São também identificadas as principais dificuldades e problemas existentes, sentidos pelos novos alunos do DEI quando se encontram pela primeira vez neste. De seguida são definidos os objetivos que visam o desenvolvimento de um protótipo com potencial para aplicar técnicas de Realidade Aumentada associadas ao processo de geolocalização para proporcionar ao utilizador a rota para chegar ao seu destino, terminando depois com a motivação do presente projeto, bem como a organização da dissertação apresentada. O segundo capítulo tem como função apresentar o contexto do tema desta dissertação. Desta forma, a secção “Realidade Aumentada” contém um estudo sobre o estado da arte, no qual será feito um levantamento do estado inicial e atual da Realidade Aumentada e da geolocalização assim como aplicações de sucesso nessas áreas. Na

secção “Contextualização Tecnológica” será feito um levantamento das tecnologias mais adequadas para a realização do projeto. Por último, na secção “Análise de Valor”, será feito uma análise do valor do negócio em que este projeto se insere, assim como os fatores que levaram à decisão final na escolha do mesmo.

No terceiro capítulo “Implementação” será realizada a análise e design da solução proposta, seguida de uma breve explicação do funcionamento da aplicação e dos seus requisitos. De seguida serão expostos os casos de uso do utilizador. De seguida será apresentado quais as soluções arquiteturais e tecnológicas pensadas e as razões que levaram à solução final. A implementação e desenvolvimento da solução serão depois descritos, desde os primeiros protótipos no Unity até à utilização do Indoor Atlas e todo o processo de mapeamento do local.

O quarto capítulo descreve como a solução será avaliada, segundo as suas funcionalidades e a satisfação do cliente. Esta avaliação permite verificar a qualquer momento o estado da solução.

O último capítulo, “Conclusão”, comporta o balanço do projeto desenvolvido e implementado ao longo do ano letivo 2019/2020. Aqui são expostos os aspetos positivos assim como as dificuldades encontradas ao longo de todo o percurso. São também expostas possíveis melhorias e implementações do projeto.

2 Contexto e Estado da Arte

“Há conhecimento de dois tipos: sabemos sobre um assunto, ou sabemos onde podemos pesquisar informação sobre ele”

Samuel Johnson

O presente capítulo tem como função apresentar o contexto do tema desta dissertação. Desta forma, a secção “Realidade Aumentada” contém um estudo sobre o estado da arte, no qual será feito um levantamento do estado inicial e atual da Realidade Aumentada e da geolocalização assim como aplicações de sucesso nessas áreas. Na secção “Contextualização Tecnológica” será feito um levantamento das tecnologias mais adequadas para a realização do projeto. Por último, na secção “Análise de Valor”, será feita uma análise do valor do negócio em que este projeto se insere, assim como os fatores que levaram à decisão final na escolha do mesmo

2.1 A Realidade Aumentada

A Realidade Aumentada é uma experiência interativa de um ambiente que simula o mundo real onde os objetos que residem no mundo são melhorados através de informação perceptual gerada por um computador. Esta melhoria pode ser feita através de múltiplas modalidades sensoriais como a visual, olfatória e auditiva.

RA pode ser definida como um sistema que cumpre três funcionalidades básicas: a combinação do mundo real e virtual, interação em tempo real e um registo preciso de objetos reais e virtuais (Wu, Lee, Chang, & Liang, 2013).

Como se pode constatar em qualquer dicionário, aumentar é o acto de acrescentar, adicionar ou enriquecer algo. No caso da Realidade Aumentada, o objetivo é aumentar o mundo real e enriquecê-lo com objetos artificiais, sem nunca se perder a noção do mundo real. A Realidade Aumentada é uma mistura de realidades entre o mundo real

e o mundo virtual, sendo uma das suas grandes vantagens, graças aos avanços tecnológicos e dos dispositivos moveis, estar a alcance dos utilizador a qualquer altura e qualquer lugar proporcionando uma experiência imersiva e de utilização intuitiva. Porém nem sempre foi este o caso. A Realidade Aumentada começou a ser falada, juntamente com a realidade virtual, em 1901 por L. Frank Baum, um autor que mencionava a ideia de um aparelho eletrónico que sobrepunha dados por cima da vida real. Com o avanço tecnológico que se obteve ao longo do século, a partir da década de 50 vários livros e documentos sobre a possibilidade da Realidade Aumentada, mas só o cinematografa Morton Heilig chegou remotamente perto, com um protótipo que intitulou de “Sensorama” que dava ao utilizador uma experiência virtual com visuais, som, cheiro e vibração (Faria & Dias, 2007) como se pode observar na figura 1.



Figura 1 – Sensorama¹

Os primeiros testes com RA datam o fim da década de 60 na Universidade de Harvard quando Ivan SutherLand publicou um artigo intitulado “The Ultimate Display” (SutherLand, 2005) em 1965 e em 1968 conseguiu juntamente com uma equipa construir o primeiro dispositivo de RA. Este dispositivo conseguia apenas renderizar as linhas principais de um sólido pois na altura os computadores não tinham capacidade

¹ Imagem disponível em https://www.researchgate.net/figure/Figura-1-Cartaz-de-divulgacao-da-Sensorama-Fonte-Telepresence-2007-Em-torno-de_fig1_305114671

de processamento para algo mais complexo. Este dispositivo consistia em monitores montados na cabeça do utilizador que o permitiam ver as imagens virtuais e os objetos reais, como se pode observar na figura 2.

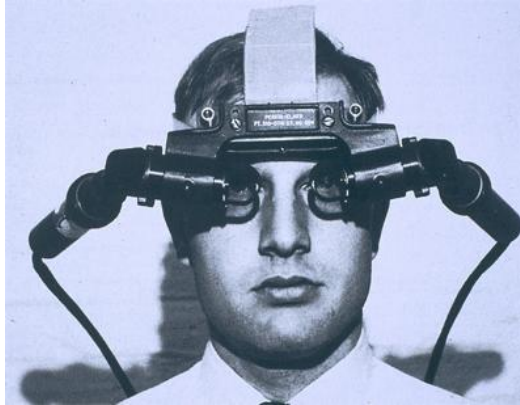


Figura 2 Primeiro protótipo de HDM, “The Ultimate Display”²

Nas décadas seguintes houve mais alguns desenvolvimentos mas só em 1992 é que surgiu mais um dispositivo que providenciava experiências imersivas de realidade mista e foi desenvolvido pelo Armstrong Laboratory da força aérea dos Estados Unidos.

Com o aparecimento do PDA apareceram projetos de RA móvel como a aplicação AR-PDA feita por Geiger e um grupo de investigadores da C-LAB em 2001 (Geiger, Kleinnjohann, & Stichling, 2001). Esta aplicação utilizava o processamento de dados do PDA para reconhecer e renderizar imagens e processar vídeos através de serviços remotos no servidor. O objetivo desta aplicação era trazer o conceito de RA para o utilizador comum e melhorar a sua rotina diária, fosse com visitas guiadas, jogos ou tarefas domésticas.

Entre 2003 e 2008, com o aparecimento dos *smartphones*, a tecnologia de RA fez a transição dos PDA para estes novos dispositivos. Porém como a capacidade de processamento dos smartphones ainda era baixa alterações tiveram de ser feitas, e então foi começado a ser utilizado o Bluetooth e a rede GSM/GPRS para comunicar com servidores que faziam o processamento e renderização da imagem (Henrysson, Billinghurst, & Ollila). Só mais tarde, entre 2005 e 2008, com o aparecimento do sdk ARToolKit, é que se conseguiram começar a criar jogos.

² Imagem disponível em <https://sensoriumresearch.wordpress.com/2012/10/15/inspiration-log-3-the-ultimate-display/>

Em 2008 também a Apple entrou no mercado com o Iphone 3G e 3GS o que possibilitou uma mudança no desenvolvimento de software de Realidade Aumentada pois inclu+iam uma nova câmara com maior resolução, bússola, ligação de dados 3G (ARToolKit, 2007) e um novo processador, o que possibilitava a utilização do ARToolKit muito mais facilmente. Também em 2011 os smartphones Android começaram a usar o ARToolKit (ARToolKit, 2007) e, mais tarde, o ARCore desenvolvido pela Google.

Anunciado em 4 de Abril de 2012, mas estando a ser trabalhado desde 2006, a Google patenteou um dispositivo chamado de Google Glass (The history of Google Glass, s.d.) que, tal como se pode ver na figura 3, era semelhante a um par de óculos mas com uma pequena tela por cima do campo de visão em um dos olhos. Este dispositivo permitia mostrar ao utilizador mapas, opções de música, atender chamadas e até mesmo apresentar rotas, permitindo ao utilizador ver estas rotas ao mesmo tempo que observava o mundo real.



Figura 3 Google Glasses ³

Em 2015 a Microsoft anuncia algo revolucionário, o HoloLens (figura 4), um capacete de Realidade Aumentada que utiliza vários sensores e uma unidade de processamento para renderizar objetos virtuais num mundo real (Microsoft, s.d.). Esta versão nunca passou de algo experimental e para âmbito empresarial visto que a Microsoft apenas vendia o capacete a parceiros que se mostrassem interessados e não ao público geral.

³ Imagem disponível em <https://www.emaze.com/@AWWTWZOQ>



Figura 4 – O Hololens da MicroSoft ⁴

Após a revelação que o Hololens não seria comercializado, várias empresas começaram a apostar em Realidade Aumentada como as diversas marcas de telemóveis que permitiram que os dispositivos mobile suportassem aplicações de Realidade Aumentada, ou como a Magic Leap que em 2017 lançou um capacete de Realidade Aumentada do mesmo tipo do Hololens mas com tecnologia mais atualizada e aberto ao público. Em 2019 a Microsoft anunciou o HoloLens 2 que traz melhorias significantes tanto ao nível ergonómico como a nível do campo de visão (Microsoft, Soluções prontas para negócios para o HoloLens 2, s.d.).

2.1.1 Realidade Aumentada Ligada à Geo Localização

Geolocalização é a identificação ou estimativa da localização geográfica de um objeto como um radar, um telemóvel ou um terminal conectado à internet. Na sua forma mais simples a geolocalização envolve a geração de um conjunto de coordenadas geográficas, mas a sua usabilidade é melhorada e utilizada atualmente como maneira de determinar uma localização significativa como por exemplo o nome de uma rua (Ionescu).

Para utilizar geolocalização é necessário um objeto para localizar e um localizador. Estes localizadores usam normalmente métodos de localização via frequências de rádio como por exemplo o Time Difference of Arrival (TDOA) para precisão. Estes métodos por normal utilizam sistemas de informação geográficos ou, caso os satélites de navegação

⁴ Imagem disponível em <https://www.windowcentral.com/microsoft-hololens>

não estejam disponíveis, utilizam as torres telefônicas de maneira a triangular uma estimativa da posição do objeto.

Atualmente as técnicas de Geolocalização associam-se muitas vezes com técnicas de RA pois permitem colocar em coordenadas geográficas específicas objetos virtuais que vão complementar o mundo real do utilizador. Alguns dos usos mais comuns são objetos informativos em locais geograficamente importantes, como por exemplo explicações de monumentos ou ratings à porta dos restaurantes. Outro uso é também a colocação de objetos orientativos de forma a guiar o utilizador em determinadas rotas.

2.1.2 Plickers (RA em Educação)

O plickers é uma ferramenta disponível em versão web e mobile para administração de testes rápidos. Esta aplicação permite a criação de testes de escolha múltipla com diferentes respostas e a impressão de cartões que depois serão lidos pela aplicação através da câmara e de forma a receber a resposta de cada aluno, como se pode ver na figura 5.



Figura 5 - Utilização do Plickers ⁵

Para tal o professor necessita de criar uma classe dentro da aplicação e de cadastrar os seus estudantes, e criar no site do plickers as questões.

O passo seguinte passa por imprimir os cartões gerados pelo sistema a cada aluno. Esses cartões tem o número de estudante, a letras das opção e um código QR. Cada um

⁵ Imagem disponível em https://miro.medium.com/max/400/0*9r96eZ8keDMaIWkt

destes códigos é diferentes para que apenas o aluno saiba qual a resposta que está a mostrar (Silva, Sales, & Castro, 2018).

Por último o professor coloca a questão aos alunos e utiliza o seu dispositivo móvel para ler a resposta de cada aluno.

De acordo com o portal Medium (Brasil, s.d.) é possível utilizar os mesmos cartões várias vezes e em diferentes turmas pois o cartão está apenas associado à letra da resposta, e ao número do aluno da turma, e não ao aluno em específico ou à questão.

Refere por isso esse portal que o uso de Plickers tem as seguintes vantagens:

- Pode ser usado mesmo quando não há muitos recursos disponíveis, pois apenas o professor precisa ter um celular com acesso à internet.
- Permite acessar com rapidez as respostas dos estudantes.

2.1.3 Story of the Forest

O museu nacional de Singapura criou uma exibição que consiste em 69 imagens do William Farquhar Collection of Natural History Drawings que foram transformadas em animações 3D que o utilizador pode interagir. Fornecem também uma aplicação para telemóvel que os utilizadores podem utilizar para apanhar plantas e animais escondidos nos quadros e verem as suas próprias coleções virtuais a crescerem, da mesma maneira que acontece com o Pokemon Go apresentado em cima. Além de ser uma maneira dinâmica de chamar a atenção destes quadros é também uma maneira didática de apresentar informação pois a aplicação mostra informação sobre cada um dos itens que foram colecionados, ensinando assim aos utilizadores factos como o habitat das espécies, a raridade destas e os seus hábitos alimentares (Singapore, 2020).

2.1.4 Skin and Bones

Em 2017 o Smithsonian Museum introduziu uma aplicação de Realidade Aumentada, Skin and Bones, às suas visitas guiadas. Esta aplicação tem a informação sobre 13 esqueletos e utiliza-a para quando um desses esqueletos aparece na camera da aplicação conseguir reconstruir essas criaturas, mostrando como é que ficaria a pele e

músculo por de cima do esqueleto, tal como se pode ver na figura 6. É possível também utilizar esta aplicação para ver o comportamento destes animais no seu habitat natural, como por exemplo, um morcego vampiro a voar (Institution, 2015).



Figura 6 - Skin and Bones⁶

2.1.5 O Google Maps

O Google Maps é um serviço de pesquisa e visualização de mapas e imagens de satélite da Terra, fornecido e desenvolvido pela Google e lançado em 2005. Este serviço modificou o conceito de mapeamento de dados utilizando recursos web devido ao fornecimento da sua ferramenta de busca por endereços, traçamento de rotas, cálculo de destinos e permissão para o uso dessa ferramenta por terceiros (Rodrigo Ramos, 2014).

Atualmente este serviço disponibiliza mapas e rotas para qualquer ponto do mundo, assim como imagens satélites com a possibilidade de zoom nas grandes cidades. Neste serviço é possível utilizar as funcionalidades GPS do dispositivo que se está a usar para informar o Google Maps da posição atual e assim descobrir qual o melhor trajeto para o destino que se pretende (Warren, s.d.).

Permite também fazer o download da maioria dos mapas e o uso offline de todas estas funcionalidades. Informa em tempo real o utilizador sobre o fluxo do trânsito, pontos

⁶ Imagem disponível em <https://www.inexhibit.com/case-studies/smithsonian-skin-bones-mobile-app-nmnh-washington/>

turísticos e nomes de ruas, assim como rotas para automóvel, transportes públicos, bicicletas e pedestres (Santos, Feitosa, & Perinotto).

Para além disso é possível em sistemas Android e iOS utilizar as funcionalidades de Realidade Aumentada para ser informado de uma forma mais intuitiva e visual das direções que terá de seguir como se pode ver na figura 7. Esta funcionalidade permite ao utilizador ter uma boa perceção do verdadeiro caminho a seguir devido aos vários indicadores visuais que vão aparecendo ao longo do seu percurso.



Figura 7 - Google Maps⁷

Outra funcionalidade de extremo interesse no google maps é o modo de Street View. Esta funcionalidade é uma representação virtual do ambiente que rodeia o utilizador, composta por milhões de imagens panorâmicas. O conteúdo apresentado pode vir diretamente da Google ou de parceiros que forneçam estas imagens, para assim ser possível oferecer aos utilizadores a possibilidade de explorar o mundo virtualmente, onde quer que estejam (Google, Street View, s.d.).

Ao entrar no Google Maps, como se pode ver na figura 8, é possível ver a visão geral de um local e do que o rodeia. Entrando depois no modo de Street View, o utilizador pode então ter uma visão 360 do ambiente que rodeia o local que selecionou. Para além disso, como se evidencia na figura 9, o utilizador é também guiado por setas indicadoras de direção que permitem escolher qual o próximo caminho a seguir, assim como se deslocar por entre a rua. Esta funcionalidade, juntamente com o modo AR do Google

⁷ Imagem disponível em <https://ispr.info/2019/05/09/google-maps-ar-navigation-makes-walking-directions-foolproof/>

Maps, foi especialmente importante pois deu novas ideias e noções de como utilizar setas direcionais em ambientes virtuais como forma de orientação e de deslocação no espaço.

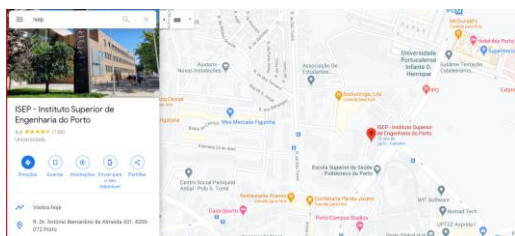


Figura 8 – Vista do ISEP pelo Google Maps



Figura 9 – Vista do ISEP através do modo de Street View

2.1.6 O Pokemon Go

O Pokemon GO é um jogo eletrónico grátis desenvolvido por uma parceria entre a Niantic, Inc., a Nintendo e a The Pokemon Company para as plataformas iOS e Android. Lançado no início de Julho de 2016 este jogo tornou-se um fenómeno global e uma das aplicações móveis mais utilizadas em 2016, tendo sido descarregada mais de 750 milhões de vezes em todo o mundo. Foi também creditado por popularizar os jogos baseados em geolocalização e Realidade Aumentada.

Ao iniciar o jogo, o utilizador cria um avatar e depois é apresentado a um mapa da sua localização (Niantic, s.d.). O utilizador facilmente consegue reconhecer este mapa como o do sítio onde se encontra, porém em várias secções do mapa pode ver objetos e

animais denominados “pokemon” que terá de se aproximar fisicamente de maneira a interagir com, pois este jogo utiliza técnicas de geolocalização para forçar ao utilizador a deslocação até vários locais, como se pode ver na figura 10.



Figura 10 – Mapa do pokemon GO⁸

Ao interagir com certos objetos e com os pokemons é também possível ligar a câmara o que leva ao segundo grande factor do jogo, a Realidade Aumentada. Ao ligar a função de Realidade Aumentada depois de interagir com um pokemon, é possível vê-lo no mundo real e ter várias interações com ele como dar-lhe de comer ou tentar apanhá-lo. Esta funcionalidade foi uma das grandes inovações do Pokemon GO pois permitia aos utilizadores realmente sentirem-se como treinadores de pokemon.

2.1.7 Conclusões sobre o estado da arte

A geolocalização é um conceito recente e extremamente útil no dia-a-dia de uma sociedade, fazendo com que a localização e escolha de trajetos seja eficiente e simples. A Realidade Aumentada é uma tecnologia recente e cada vez mais em desenvolvimento e com um progresso em pique, sendo que agora é o momento ideal para começar a desenvolver aplicações com base nesta tecnologia.

⁸ Imagem disponível em <https://www.nintendoblast.com.br/2017/12/pokemon-go-guia-clima-dinamico.html>

Já há várias empresas com soluções que conjugam a geolocalização e Realidade Aumentada, mas nenhuma com o intuito de localizar e orientar utilizadores em espaços fechados.

2.2 Contextualização Tecnológica

De forma a criar uma aplicação em Realidade Aumentada que guie o utilizador é necessário ter em conta tres aspectos tecnológicos que irão modificar a forma de obtenção da solução: Como se irá fazer a programação para dispositivos móveis, como se irá fazer a Realidade Aumentada, e como é que se consegue obter fielmente a localização do utilizador e qual o caminho a seguir.

2.2.1 O Android Studio

A maneira mais simples de começar a fazer uma aplicação para dispositivos mobile android é utilizando a ferramenta Android Studio.

Android Studio é um ambiente de desenvolvimento integrado, IDE, lançado pela primeira vez Maio de 2013, mas a primeira versão estável saiu em Dezembro de 2015 (Protalinski, 2014). Foi feito especificamente para desenvolvimento em android, estando disponivel em dispositivos Windows, Mac OS X e Linux, e é o IDE principal da Google para desenvolvimento nativo para Android (Gerber & Craig).

O Android Studio tem suporte para compilações baseadas em Gradle, suporte nativo para a Google Cloud Platform, refactoring especifico para Android, um editor de layout rico que permito aos utilizadores construirem a sua user interface de forma simples e intuitiva. Para além disso trabalha com multiplas linguagens como Java, C++ e Kotlin, tem facil integração com diversas bibliotecas e SDK e permite utilizar tanto dispositivos virtuais como fisicos para compilar e depurar (Developers A. , s.d.).

Ao contrário dos motores de jogo, que irão ser falados a seguir, o Android Studio não tem um motor gráfico nem uma abstração das chamadas de renderização, portanto qualquer interação com bibliotecas de renderização terá de ser controlada manualmente pelo utilizador.

2.2.2 Motores de jogo

Um motor de jogo é um programa que agrupa um conjunto de bibliotecas de forma a simplificar e abstrair o desenvolvimento de jogos eletrônicos ou aplicações com gráficos em tempo real. A funcionalidade tipicamente fornecida por um motor de jogo inclui um motor gráfico para renderizar gráficos 2D e/ou 3D, um motor de física para detecção de colisões e físicas dos objetos, suporte de animação, som, inteligência artificial, *networking*, gestão de memória, suporte de uma linguagem de programação, entre outros (Gregory). Embora existam vários motores de jogo, os dois mais conhecidos atualmente são o Unreal e o Unity.

2.2.2.1 O Unreal Engine

O Unreal Engine (figura 11) é um motor de jogos e IDE criado pela Epic Games. Foi lançado a primeira vez em 1998, juntamente com o jogo Unreal, e continua renderização, detecção de colisões, sistema de redes e de arquivo e algumas bibliotecas de inteligência artificial. Utilizava a API Glide ao invés de OpenGL, que era o mais normal de se utilizar na altura, e continha uma linguagem de script, o que tornou esta engine bastante popular na altura.



Figura 11 - Logotipo do unreal Engine⁹

Atualmente o Unreal Engine vai na quarta versão, Unreal Engine 4. Foi lançado globalmente em 19 de Março de 2014, tendo updates regulares de manutenção e de lançamento de novas features como por exemplo o mercado que permite os utilizadores comprarem e venderem conteúdos criados pela comunidade.

⁹ Imagem disponível em <https://www.unrealengine.com/en-US/>

Além da sua popularidade, este motor possui também uma série de prêmios e jogos de sucesso que foram desenvolvidos com a ajuda desta. Recebeu o título de melhor motor de jogo do ano na Develop Industry Excellence Awards nove vezes de 2008 a 2018, não sendo vencedor apenas em 2014 e 2015 (Games, Awards, s.d.). Tem também vários jogos premiados nas The Game Awards, onde se pode destacar os premiados: Injustice 2, PlayerUnknown's Battlegrounds, What Remains of Edith Finch, Fortnite entre outros (Games, Senua's Saga: Hellblade II highlights a long list of Unreal Engine titles at The Game Awards 2019, 2019).

Outro fator que aumenta a popularidade e uso deste motor de jogo é o facto de disponibilizar todos os seus recursos, desde o próprio motor, a variados assets que vão desde texturas, a modelos 3D e a ficheiros de som utilizados tanto em tutoriais como em alguns jogos da autoria deles como por exemplo o Paragon. Tudo porém requer que caso os produtos desenvolvidos sejam comercializados, seja pago à Unreal 5% dos ganhos, Não contabilizando os primeiros três mil dólares de cada trimestre para cada produto e os primeiros cinco milhões de dólares para cada produto na Oculus Store (Games, Unreal® Engine End User License Agreement For Publishing, s.d.).

A versão mais atual, v4.25, conta com suporte às novas gerações de consolas, integração de pipelines, um conceito de programação visual recorrendo ao uso de blueprints, integração com suporte audio-visual, suporte de realidade virtual e aumentada, editor de meshes, entre muitas outras funcionalidades (Games, Unreal, s.d.). Possibilita também a utilização de scripts em C++.

2.2.2.2 O Unity

O Unity (figura 12) é um motor de jogos e IDE criado pela Unity Technologies, lançado pela primeira vez em 8 de junho de 2005. Esta versão era limitada a Mac OS, e continha um renderizador OpenGL focado para shaders, um motor de fisica, suporte de audio e integração com código em C#.

Várias versões e updates foram sido feitos ao longo dos anos e atualmente o Unity vai na versão Unity 2020, tendo sido lançada em alpha a setembro de 2019 e finalizada a julho de 2020.

Para além da popularidade que foi adquirindo ao longo dos anos devido às várias funcionalidades introduzidas em cada update, este motor de jogo ficou principalmente conhecido devido aos títulos desenvolvidos dentro desta, como por exemplo o jogo Cuphead, que foi vencedor das categorias de melhor direção de arte e melhor jogo indie, nas The Game Awards 2017 (Awards, 2017).

Outro factor que popularizou esta engine é o facto desta ser gratuita com todas as suas principais funcionalidades para qualquer utilizador particular, ou empresa com receita bruta anual abaixo dos 100 mil dólares (Unity, Store, s.d.).

A última versão lançada conta com um package manager para gerir os vários plugins oficiais lançados, como por exemplo o google ARCore, uma loja de assets para comprar e vender criações da comunidade, uma pipeline de renderização em alta definição, suporte para elementos audio-visuais e sistemas de partículas, suporte para realidade virtual e aumentada, assim como suporte para todas as consolas atuais e dispositivos móveis, e suporte para criação de ambientes 2D e 3D. Neste momento o Unity suporta nativamente a linguagem de programação C#. Muitas outras linguagens de .NET podem também ser utilizadas em conjunto com o unity, desde que compiladas numa DLL compatível (Unity, Products, 2017).



Figura 12 – Logotipo do Unity¹⁰

2.2.3 SDK de Realidade Aumentada

Embora os motores de jogo facilitem bastante a construção de aplicações gráficas para dispositivos móveis, estes, por norma, não têm acesso às bibliotecas necessárias para renderizar objetos virtuais na câmara do utilizador, necessitando por isso de ajuda de

¹⁰ Imagem disponível em <https://unity3d.com/pt/unity>

um elemento externo, ao qual denominamos de SDK, que é de certa forma uma biblioteca que o motor de jogo, ou até mesmo uma aplicação feita de raiz sem nenhum motor por de trás, vai chamar de forma renderizar os elementos virtuais. Existem vários SDK para este efeito mas de seguida serão apresentados os mais completos para tal.

2.2.3.1 O WIKITUDE

Wikitude foi a primeira empresa a lançar uma aplicação móvel de Realidade Aumentada, em 2008. Depois disso embarcaram numa jornada para levar a realidade aumentada a novos níveis de performance e capacidade. De maneira a conseguirem isso, lançaram um SDK poderoso que inclui várias funcionalidades fundamentais como o suporte para localização e mapeamento, providencia reconhecimento 3D, reconhecimento de imagem, reconhecimento de nuvens de pontos, reconhecimento de cilindros, integração com serviços de geolocalização, integração com Unity, entre outros. Para além disso suporta desenvolvimento para Android, iOS, Windows e smart glasses (Wikitude, s.d.).

2.2.3.2 O MAXST

Desde 2010 que a MAXST tem sido uma empresa sempre na atualidade no que diz respeito à pesquisa e desenvolvimento no âmbito da Realidade Aumentada. Sendo a maior empresa de Realidade Aumentada na Coreia, criaram o MAXST AR SDK que providencia várias funcionalidades como o mapeamento de ambientes, reconhecimento de imagens, marcadores, códigos QR, códigos de barras e objetos. Para além disso permite também guardar ficheiros que serão posteriormente renderizados em 3D, e posicionar objetos virtuais em relação a planos reais. Permite também desenvolver para iOS, Android e Windows, porém exige uma quantia monetária de 600 dolares anuais caso a aplicação que utilize o SDK seja comercializada (MAXST, s.d.).

2.2.3.3 O VUFORIA

Vuforia é um sdk desenvolvido pela PTC, que já tem um longo historial no que toca a desenvolvimento de tecnologias de Realidade Aumentada. Este SDK contém várias funcionalidades das quais se destacam o reconhecimento de cilindros, imagens,

marcadores, áreas e objetos modelados previamente, planos, acesso a uma câmara externa que não a integrada no dispositivo móvel, reconhecimento de texto e armazenamento na nuvem. Contém também um plugin para o Unity e suporta desenvolvimento para iOS, Android e Windows. A sua utilização requer no entanto um custo de 42 dolares anuais caso o produto seja comercializado, que acresce para 99 dolares caso seja requisitado o uso da nuvem. É possível pedir também uma licença académica de 500 licenças pelo valor de 3000 dolares anuais (Vuforia Engine, s.d.).

2.2.3.4 O ARTOOLKIT

O ARToolKit é uma biblioteca open-source dedicada à criação de aplicações de Realidade Aumentada. Embora seja uma biblioteca grátis e construída pela comunidade, providencia uma variedade de funcionalidades como calibração de câmara, reconhecimento de marcadores quadrados, rapidez e eficiência para aplicações de realidade aumentada em tempo real e deteção da orientação e posicionamento da câmara. Tem também suporte para smart glasses, aplicações em Unity e permite a distribuição em sistemas SGI IRIX, Linux, MacOS e Windows OS (ARToolKit, 2007).

2.2.3.5 O GOOGLE ARCORE

O ARCore da Google é um dos SDK mais conhecidos para desenvolvimento em Android e dos mais utilizados, seja nativamente ou através de frameworks criadas para facilitar o uso deste SDK e da sua interação com a componente gráfica da aplicação.

Este SDK conta com rastreamento de movimento, determinando assim a posição e orientação do dispositivo e encontra pontos específicos do ambiente para assim posicionar corretamente objetos virtuais. Para além disso tem funcionalidades de conhecimento ambiental localizando assim superfícies horizontais como mesas e o chão para posicionamento de objetos virtuais. Contém também funcionalidades de estimação de luz de maneira a que a aplicação possa coincidir a luz virtual dos objetos com a luz real do ambiente de maneira a que estes pareçam naturais em relação ao espaço que os rodeia. Para além disso tem integração com Unity, Unreal e OpenGL, porém como é desenvolvido pela google, é especializado para o desenvolvimento em Android, não funcionando por isso para iOS nem Windows (Developers G. , ARCore, s.d.).

2.2.3.6 Comparação entre SDK

Tens agora já alguma noção de alguns dos mais conhecidos SDK para Realidade Aumentada, é possível fazer uma tabela comparativa, como a da tabela X, tendo em conta parâmetros como o custo de uso, integração com Unity, deteção de objetos, utilização de marcadores e plataformas acessíveis.

Tabela 1 – Comparação entre SDK de AR

	Custo	Unity(3D)	Marcadores	Deteção de objetos	Plataformas
Wikitude	Caso seja comercializado	sim	sim	sim	Android, iOS, Windows e smart glasses
MaxST	Caso seja comercializado	Sim	sim	sim	iOS, Android e Windows
Vuforia	Caso seja comercializado	sim	sim	Pequenos objetos	iOS, Android e Windows
ARToolKit	Grátis	Sim	Sim	não	iOS, Android e Windows
ARCore	Grátis	Sim	Sim	sim	Android

Através desta comparação é possível perceber que embora a maioria dos SDK estejam em pé de igualdade quanto a funcionalidades, só o ARToolKit e o ARCore é que disponibilizam todas as suas funcionalidades gratuitamente, sendo por isso os dois SDK que serão explorados no decorrer da solução.

2.2.4 Tecnologias de orientação no espaço

De maneira a conseguir orientar o utilizador pelo espaço é necessário utilizar técnicas para descobrir a localização específica do utilizador a qualquer momento e qual o caminho que ele terá de fazer para chegar ao local desejado. De seguida será apresentado algumas das tecnologias que se aplicam a este problema, assim como os seus pontos positivos e negativos.

2.2.4.1 A Google Geolocation API

Desenvolvida pela Google, esta API permite saber a localização em coordenadas geográficas do utilizador usando a informação fornecida por torres de telecomunicações ou pontos de Wi-fi.. Para utilizar a API é necessário primeiramente ter uma chave de acesso que é pedida com uma conta Google e a comunicação é feita através de HTTPS usando POST. Tanto os pedidos como as respostas são formatados em JSON (Developers G. , Overview, s.d.).

Embora esta API seja poderosa, principalmente se usada juntamente com os mapas fornecidos pela google, tem o problema de não ter o rigor suficiente para a orientação dentro de espaços fechados nem o mapeamento dos ditos espaços, sendo assim impossível ser utilizada para encontrar o caminho ideal.

2.2.4.2 O Pedometro e o Giroscópio

A maneira mais básica de se conseguir orientar no espaço com o telemóvel é através do pedometro e do giroscópio. Integrados em praticamente todos os smartphones atuais, o pedometro conta o número de passos que o utilizador deu. Por outro lado o giroscópio integrado do smartphone permite saber a orientação atual do dispositivo (Passaro, Cuccovillo, & Vaiani, 2017). Em teoria, o conjunto destes dois elementos é o suficiente para, sabendo a posição inicial do utilizador, saber em que direção ele andou e qual a distância (Clemes & Biddle). Sabendo esta distância e tendo mapeado as distâncias e direções que o utilizador tem de percorrer para chegar ao seu destino, é possível orientar corretamente o utilizador. Na prática este sistema é pouco viável pois mesmo fazendo uma calibração prévia do pedometro, o erro deste é considerável se o

utilizador mudar um pouco o seu tipo de passada. Para além disso para saber a direção em que o utilizador se está a movimentar é necessário que o dispositivo se encontre na mesma direção que o andamento do utilizador, o que poderá não ser o caso em diversos cenários.

2.2.4.3 Here Indoor Positioning API

Desenvolvido pela Here, este SDK permite utilizar os pontos de Wi-fi. e de Bluetooth como Beacons para descobrir corretamente a posição do utilizador. Pode ser utilizado em aplicações nativas para Android e IOS embora para IOS não possa ser utilizado a rede de Wi-fi. Para além disso, de forma a permitir a orientação em edifícios fechados é necessário pedir arranjar uma licença Here Public Indoor Positioning e pedir a um representante da Here para fazer o mapeamento do edifício. Pode-se também utilizar um mapa pessoal customizado, porém é preciso passos extra para o alinhar com o mapa fornecido pela API e não funciona com o resto das funcionalidades do SDK como por exemplo encontrar o caminho mais rápido para um certo local (Developer, s.d.).

2.2.4.4 Indoor Atlas SDK

O SDK da IndoorAtlas permite localizar o utilizador através de sinais de Wi-fi, Bluetooth, o campo magnético, e a pressão atmosférica usando o barómetro do telemóvel. Permite também utilizar plantas próprias e transmitir ao SDK quais os pontos do mapa que podem ser visitados e quais os caminhos possíveis. Depois de alinhar o mapa fornecido com o mapa do Indoor Atlas, e mapear os locais e caminhos, o SDK calcula automaticamente qual o caminho mais curto entre a posição do utilizador e o destino. Este SDK é de uso grátis e funciona para aplicações nativas Android e IOS e tem integração com o Unity. Porém para otimizar todo o cálculo do percurso é necessário que a planta tenha um rigor bastante profissional, e que seja fornecido também a posição dos pontos de Wi-fi e/ou dos beacons. Se estes não existirem, a posição encontrada através do campo magnético poderá ter um erro entre 1.5 e 2 metros (Atlas, s.d.).

2.2.5 Conclusões sobre as tecnologias

Após a consideração destas várias tecnologias, foi decidido numa primeira abordagem a utilização de um motor de jogo em vez de fazer uma aplicação nativa devido a ser uma aplicação com um grande foco na renderização e no ambiente gráfico. Dentro das duas estudadas foi decidido o uso do Unity pois o aluno estava mais confortável com este motor e por ter uma maior integração com todos os SDK referidos, tanto a nível de Realidade Aumentada como de geolocalização. Porém, devido a motivos que irão ser apresentados no “Implementação”, numa segunda fase foi necessário substituir o Unity pelo Android Studio.

Quanto aos SDK utilizados foram escolhidos o Google ArCore e o ARToolKit devido ao vasto conteúdo que providenciam, o facto de não terem custos, e terem uma fácil integração tanto com o Unity como com o Android Studio.

2.3 Análise de valor

A principal razão para a criação de um produto é cumprir as necessidades do cliente. Desta forma essas necessidades irão se tornar em funcionalidades que o produto terá de ter. De forma a perceber como satisfazer ao máximo as necessidades do cliente é necessário realizar uma análise de valor.

A análise de valor é um processo sistemático, formal e organizado de análise e avaliação e tem como objetivo estudar quais as melhores formas de aumentar o valor de um produto pelo menor custo possível, sem sacrificar qualidade, confiabilidade, desempenho e aspeto (Library, 2015).

O valor pode ser calculado através da seguinte fórmula:

$$Valor = \frac{Benefícios}{Custo}$$

Figura 13 - Fórmula do valor

Olhando para a equação percebe-se que para aumentar o valor do produto é necessário diminuir custos mantendo os benefícios existentes ou mantendo os custos enquanto se aumenta os benefícios.

Para além do visto em cima, o valor de um produto é avaliado na análise de valor através de duas perspetivas. Estas perspetivas são:

- **Valor Utilitário:** Valor dado pela funcionalidade e utilidade do produto;
- **Valor para o Cliente:** Valor que o cliente dá aos atributos do produto que não contribuem diretamente para a utilidade do produto, como por exemplo, o aspeto gráfico da aplicação.

Tendo em conta estas duas perspetivas é possível calcular o **Valor de Mercado** para o produto como pode ser entendido pela figura 14.

$$\text{Valor de Mercado} = \text{Valor Utilitário} + \text{Valor para o Cliente}$$

Figura 14 - Fórmula do valor de mercado

Tendo em conta as definições de valor e fórmulas mencionadas anteriormente, pode-se definir que valor é traduzido no equilíbrio entre a satisfação do cliente e o preço. O conceito de valor vai de encontro ao princípio em que o consumidor procura sempre o melhor produto ao menor preço.

Por outro lado, o cliente não tem a perceção de todos os benefícios que um produto ou serviço pode fornecer, por isso a sua perceção do valor não corresponde, por norma, à realidade do produto.

De maneira a informar o cliente do valor real de um produto, muitas empresas recorrem ao marketing. O marketing servirá então não só para divulgar o produto, como também para informar o cliente sobre as características do mesmo, aumentando a perceção de valor relativo ao produto.

Recorrendo à figura 12, pode-se afirmar que se o valor for superior a 1, o produto tem um valor justo ou bom. Ou seja, os benefícios são superiores aos custos, o que por sua vez se traduz num bom investimento para o cliente.

Pelo contrário, se o valor é inferior a 1, ele tem um valor baixo. Ou seja, os custos são superiores aos benefícios, o que leva o cliente a considerar apostar no produto um mau investimento.

De forma a analisar melhor o valor percebido pelo cliente, é realizada uma tabela dos benefícios e sacrifícios em relação ao produto, serviço e à relação. Isto permite também validar se a percepção que a empresa tem do valor percebido pelo cliente está correta. Na tabela 2 pode-se observar os benefícios e sacrifícios da aplicação.

Tabela 2- Benefícios e sacrifícios

	Produto	Serviço	Relação
Benefícios	Soluções Alternativas Qualidade Personalização	Responsividade Flexibilidade Confiabilidade Competência Técnica	Imagem Confiança Solidariedade
Sacrifícios	Preço		Tempo Esforço

2.3.1 Processo de Inovação

A inovação, em qualquer área, é fundamental para o bom desenvolvimento de um produto e traz bastantes vantagens competitivas, porém, acarreta também diversos riscos.

De forma a mitigar os riscos, técnicas para o processo de inovação estão constantemente a serem desenvolvidas. Uma das mais utilizadas é o modelo de Peter Koen.

Peter Koen divide o processo de inovação em três fases distintas: *Fuzzy Front End* (FFE), *New Product Development* (NPD) e Comercialização. O modelo pode ser observado na figura 15.

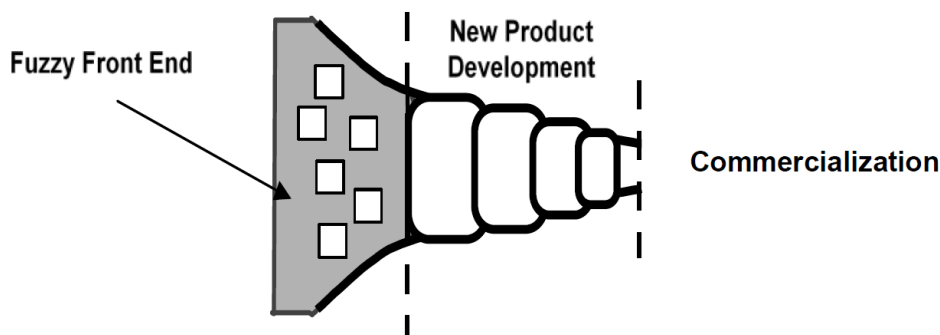


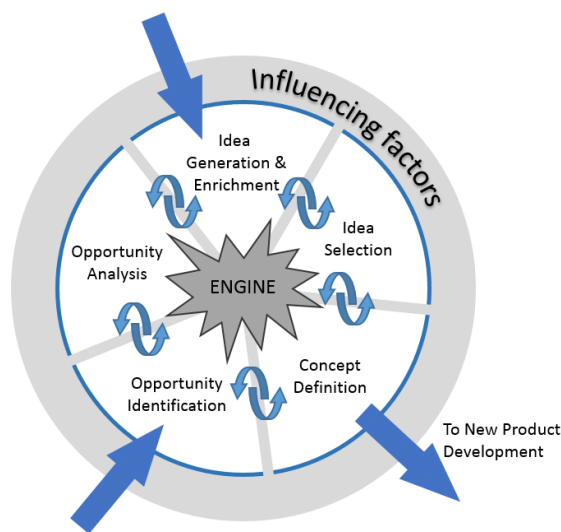
Figura 15 – Processo de inovação do produto

O FFE é caracterizado pela informalidade, atividades imprevisíveis e desorganizadas e datas incertas. Na segunda fase existe um grau maior de formalização, os objetivos são realmente definidos e o foco para o seu cumprimento aumenta e datas rigorosas para a conclusão das tarefas são definidas. Por último tem-se a parte final que é a comercialização do produto (Koen, 2002).

De forma a dar ao FFE uma linguagem e vocabulário comum, foi desenvolvido em 2001 o *New Concept Development* (NCD) (Martikainen, 2017).

O NCD é composto por três partes essenciais como podemos ver na figura 16.

- **Engine:** que representa a liderança, a cultura e estratégia empresarial da organização;
- **Fatores influenciadores:** são os fatores externos que influenciam o engine e os elementos chave;
- **Elementos Chave:** nesta parte enquadra-se a identificação e análise da oportunidade, geração e análise da ideia, e conceito e desenvolvimento tecnológico; como é possível ver na figura 16, é possível passar de um elemento para outro diversas vezes.



11

Figura 16 – New Concept Development Model

2.3.1.1 Identificação da Oportunidade

Neste elemento, são identificadas as variadas oportunidades a que o produto vai dar resposta. De forma a tornar a identificação de oportunidades mais eficaz, existem diversos métodos que podem ser usados, como por exemplo a análise das tendências do mercado, a análise das tendências das tecnologias e *brainstorms*.

Foi detetado que o processo de guia dentro de um edifício fechado, como por exemplo no edifício B do ISEP, ainda é feito manualmente através de mapas ou guias. Sendo assim existe a oportunidade de automatizar este processo.

2.3.1.2 Análise da Oportunidade

As oportunidades detetadas anteriormente deverão ser analisadas e avaliadas para determinar a sua viabilidade e criar um produto.

Ao criar uma aplicação que guie o utilizador pelo DEI é possível aumentar a sua satisfação e poupar em custos como mapas dentro do edifício, sinalizações ou guias.

¹¹ Imagem disponível em https://www.researchgate.net/figure/The-New-Concept-Development-NCD-model-Koen-et-al-2001_fig12_324208591.

2.3.1.3 Geração de Ideias

Este elemento consiste em utilizar a oportunidade anteriormente detetada e formular ideias concretas até chegar a um nível de detalhe capaz de ser executado.

Através de um brainstorming foi desenvolvida a ideia de criar uma aplicação que guie o utilizador a partir do momento em que ele entra no edifício, mostrando todos os locais que pode visitar e guiando-o até estes.

2.3.1.4 Seleção de Ideias

Na maioria dos negócios diversas ideias são geradas, fazendo com que seja crítico seleccionar as ideias que tragam o máximo de valor ao negócio.

As ideias que foram geradas foram filtradas de forma a obter um produto que traga ao cliente a maior quantidade de informação possível, mas de maneira a que ele a receba de uma forma agradável e intuitiva.

2.3.1.5 Conceito e Desenvolvimento Tecnológico

O último elemento do modelo consiste no desenvolvimento do negócio através das ideias seleccionadas anteriormente. Sendo que este modelo é um modelo não linear, este elemento pode na verdade ser o primeiro.

De forma a desenvolver o produto final, o desenvolvimento foi dividido em três fases:

- **Mapeamento do ambiente:** Esta fase será responsável pelo mapeamento do edifício na nossa aplicação de maneira a que se possa guiar o utilizador. Será o foco principal do projeto.
- **Integração com o Google Maps:** esta fase será a responsável por interligar a aplicação com o Google Maps e sobrepor-lhe a planta do edifício, para que o utilizador possa ter uma noção de onde se encontra e do plano geral do edifício.
- **Extras:** encarregue de criar as funcionalidades adicionais não obrigatórias para ter um produto mínimo viável, como por exemplo a integração da aplicação com a aplicação de chamadas do telemóvel para ligar aos serviços de informação do edifício, integração com Markers, ou um serviço de mensagens dentro da aplicação.

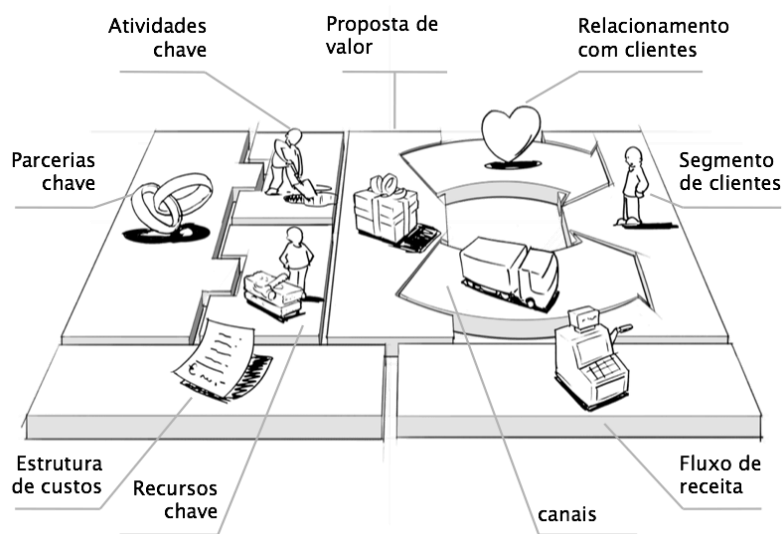
2.3.2 Quadro de Modelo de Negócio

O Quadro de Modelo de Negócio, ou em inglês *Business Model Canvas*, é uma ferramenta de gestão estratégica que é normalmente utilizada para representar o modelo de negócio de uma empresa (NOS, s.d.).

Esta ferramenta divide o modelo de negócio em 9 blocos:

- Atividades chave;
- Proposta de valor;
- Relacionamento com clientes;
- Segmento de Clientes;
- Fluxo de receita;
- Canais;
- Recursos Chave;
- Estrutura de Custos;
- Parcerias chave.

Cada um destes blocos interliga-se de variadas maneiras, como se pode observar na figura 17. Sendo que a empresa deseja transmitir a proposta de valor ao segmento desejado de clientes, este relaciona-se diretamente com a distribuição do produto e o relacionamento dos clientes, como se pode perceber através do lado direito da figura. No lado esquerdo tem-se as atividades, recursos e parceiros chave que serão essenciais para a empresa cumprir a proposta de valor. Por último pode-se observar em baixo a estrutura de custos e fontes de receita que são a base de qualquer negócio (Pereira, 2016).



12

Figura 17 – Estrutura do *Business Model Canvas*

Na tabela 3 está apresentado o quadro do modelo de negócio relativo à aplicação.

Tabela 3 – Quadro de Modelo de Negócio

<u>Parceiros Chave</u>	<u>Atividades Chave</u>	<u>Proposta de Valor</u>	<u>Relação com os Clientes</u>	<u>Segmentos de Mercado</u>
<ul style="list-style-type: none"> Serviços de geo-localização Fabricantes de telemóveis 	<ul style="list-style-type: none"> Desenvolvimento e respetiva manutenção do produto; Mapeamento dos locais; 	<ul style="list-style-type: none"> Aplicação em realidade virtual para localização dentro do Departamento de Engenharia Informática do ISEP 	<ul style="list-style-type: none"> - Contacto de e-mail de apoio dedicado. Realizadas pelas próprias plataformas de distribuição de 	<ul style="list-style-type: none"> Ensino Turismo Gestão de eventos

¹² Imagem disponível em: <https://analistamodelosdenegocios.com.br/o-que-e-o-business-model-canvas/>

			aplicações móveis;	
	<u>Recursos Chave</u> <ul style="list-style-type: none"> • Plantas do edifício; • Software de Realidade Aumentada; • Desenvolvedores de Realidade Aumentada; • Hardware; • Markers; 		<u>Canais</u> <ul style="list-style-type: none"> • Plataformas de distribuição de aplicações móveis • Eventos e convenções; 	
<u>Estrutura de Custos</u> <ul style="list-style-type: none"> • Desenvolvimento da solução; • Manutenção do sistema; • Licenças de software; 			<u>Fontes de Receita</u> <ul style="list-style-type: none"> • Venda da solução e manutenção; 	

2.3.2.1 Parceiros Chave

Os parceiros chave são entidades externas que executam atividades chave ou fornecem recursos chave para o negócio.

Os parceiros chave seriam os serviços de geolocalização pois poderiam criar formas de se tornar mais fácil o mapeamento em pequenos espaços e os fabricantes de telemóveis pois a Realidade Aumentada em dispositivos móveis ainda não é uma tecnologia dada por garantida em todos os dispositivos.

2.3.2.2 Atividades Chave

As atividades chave são as atividades que o negócio deve realizar de forma a funcionar corretamente e criar o valor proposto.

No caso deste projeto, as atividades chave são o mapeamento do ambiente a utilizar e de seguida o desenvolvimento da aplicação e a sua manutenção.

2.3.2.3 Recursos Chave

Os recursos chave são os recursos indispensáveis para a realização das atividades chave e para a criação do valor proposto.

De forma a executar as atividades anteriormente mencionadas são necessários diversos recursos, como por exemplo as plantas do edifício, Markers para colocar informação adicional, software próprio para desenvolver com Realidade Aumentada, hardware próprio e desenvolvedores experientes no ramo da Realidade Aumentada.

2.3.2.4 Proposta de Valor

A proposta de valor descreve o valor que o negócio irá disponibilizar ao segmento de mercado específico.

O desenvolvimento de uma solução para guiar e localizar utilizadores dentro do DEI, dando-lhes o máximo de informação útil possível de forma simples e intuitiva.

2.3.2.5 Relação com os Clientes

Descreve como será feita a comunicação entre o negócio e os seus clientes.

A aplicação terá por base um cliente chave, o dono do estabelecimento. O contacto com este será feito através de eventos de exposição do produto, através de emails, ou pessoalmente. Posteriormente se o dono do estabelecimento decidir disponibilizar a aplicação aos seus clientes esta comunicação poderá ser feita através dos serviços de distribuição de aplicações mobile. No caso específico desta prova de conceito, o cliente chave será o DEI ISEP.

2.3.2.6 Canais

Os canais descrevem como o produto irá chegar aos clientes;

Em termos de divulgação do produto este irá ser divulgado aos donos de grandes estabelecimentos através de eventos e convenções do ramo. Quanto aos clientes desse estabelecimento a divulgação passará por publicidade dentro do estabelecimento e através dos próprios canais deste.

Para a distribuição do produto esta poderá ser feita através dos canais já adequados para distribuição de aplicações mobile

2.3.2.7 Segmentos de Mercado

Os segmentos de mercado são os grupos de clientes a que o produto se destina. A proposta de valor e outros elementos serão desenvolvidos tendo em vista criar o máximo de valor para o segmento de mercado selecionado.

Este produto destina-se ao ISEP dado que é o dono do edifício, mas também a qualquer empresa que queira utilizar o edifício, portanto poderá singrar no ramo do ensino, turismo e gestão de eventos.

2.3.2.8 Estrutura de Custos

A estrutura de custos é o conjunto de custos necessários para desenvolver e manter o negócio.

Durante o desenvolvimento da solução irá existir custos com salários e outras despesas de pessoal, custos com o hardware.

Posteriormente irá ter custos relacionados com a manutenção do sistema e com licenças de software de terceiros.

Haverá outros custos que devem ser suportados para garantir o correto funcionamento das atividades.

2.3.2.9 Fontes de Receita

As fontes de receita descrevem de que maneira o negócio vai obter receita e gerar lucro. Esta receita poderá ser feita através de um preço inicial de venda, seguido por um custo anual de manutenção do produto.

2.3.3 QFD

O QFD (Quality Function Deployment) é um sistema para planejar um produto ou serviço baseado nos requisitos do cliente e envolve todos os membros da organização. Este sistema prevê que as responsabilidades para produzir um item de qualidade tem de ser atribuídas a todas as partes da organização.

Os propósitos do QFD focam-se em três partes. Primeiro permite obter produtos de maior qualidade mais rápido no mercado e por um menor custo. Em segundo lugar permite que se consiga atingir um produto que é criado especificamente para o cliente e por último providencia uma traçagem do sistema para futuros melhoramentos (Group).

O QFD consegue atingir estes propósitos partindo os requisitos dos clientes em segmentos e identificando meios necessários para os atingir. Os processos do QFD são os seguintes.

2.3.3.1 Requisitos de cliente

Qualquer empresa para se manter ativa tem de vender os seus produtos ou serviços e poder contar com que esses clientes se mantenham. Para que isso aconteça é necessário conhecer os requisitos dos clientes.

Porém para conhecer os requisitos dos clientes é necessário que primeiro se perceba o que é um cliente no contexto do QFD e perceber qual é a Customer Chain.

O cliente no QFD é toda e qualquer pessoa, externa à empresa, que compre o seu produto, compre o produto de uma empresa concorrente, trocaram da empresa para uma concorrente, e também todos aqueles que estão satisfeitos atualmente com o produto e todos aqueles que não estão (Group) .

Customer Chain é o conjunto de clientes que realmente vão comprar um produto, por exemplo um cereal para crianças terá como primeiro cliente o supermercado que por sua vez os vai vender aos pais que por sua vez vão dar os cereais ao cliente final, a criança.

Sabendo qual o cliente é preciso recolher os requisitos destes através de questionários e ou entrevistas.

No caso do projeto a desenvolver, a Customer Chain é constituída por dois clientes: o dono do estabelecimento e depois os clientes desse estabelecimento. Os requisitos podem ser divididos em duas partes:

Dono do estabelecimento:

- Tem de ser possível distribuir a aplicação facilmente;
- Tem de ser possível adicionar/atualizar informações;
- Tem de ser possível mandar notificações informativas aos utilizadores.

Cliente do estabelecimento:

- A aplicação tem de ser responsiva;
- A aplicação tem de ter uma interface agradável;
- Tem de ser possível verificar todos os locais acessíveis.

2.3.3.2 Requisitos Técnicos

Depois de se definir os requisitos do cliente, ou seja, aquilo que a aplicação terá de ter, é necessário definir os requisitos técnicos, ou seja, como é que se vai fazer a aplicação para cumprir os requisitos do cliente. Estes requisitos devem descrever o produto em termos medíveis e deve afetar diretamente a perceção do cliente.

No caso da aplicação em estudo os seguintes requisitos técnicos serão necessários:

- O tamanho útil de ecrã que a aplicação usa.
- Geração de um executável.
- Geração de markers personalizados.
- Sistema de notificações.
- Mapeamento de todo o local para uma estrutura de dados.
- Listagem de todas as localizações mapeadas.
- Sistema reativo.
- Tamanho dos widgets.
- Cor dos widgets.

2.3.3.3 Valores Alvo

Por último é necessário determinar valores alvo para os requisitos técnicos. Estes valores alvo refletem apenas o que é necessário para garantir que o cliente fica satisfeito, não tem em relação o que pode ou não ser atingido atualmente.

No caso da aplicação em estudo, podem ser considerados valores alvo por exemplo a geração de um executável com menos de peso menos de um gigabyte, ou que os markers tenham 5cm^2 de área.

2.3.3.4 House of Quality

Após os requisitos do cliente, requisitos técnicos e valores alvo serem definidos é possível criar uma House of Quality como se pode observar na figura 18. House of Quality é uma tabela que relaciona os requisitos do cliente com os requisitos técnicos sendo assim possível perceber que requisitos técnicos são mais fulcrais para que o cliente fique satisfeito e também perceber como os requisitos se relacionam entre si.

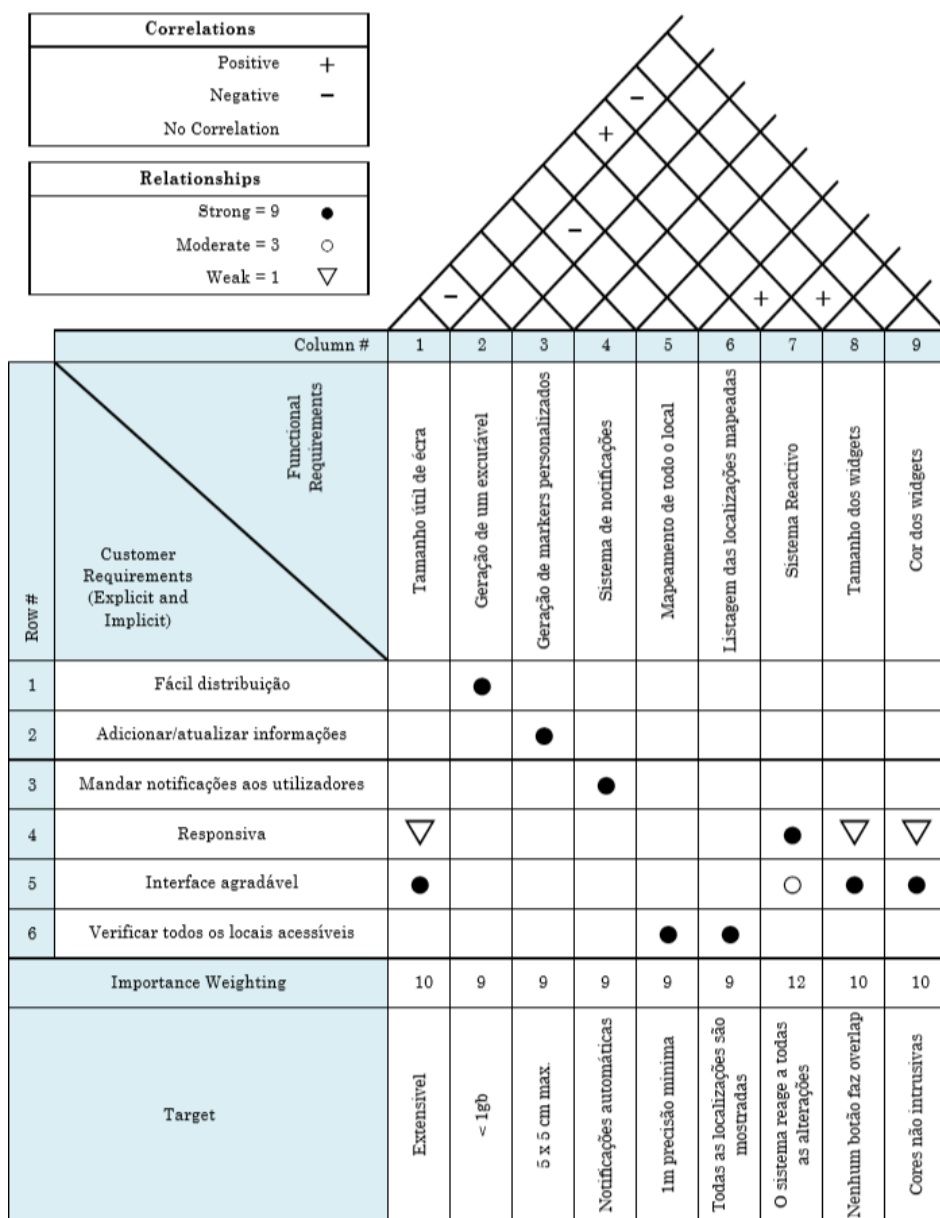


Figura 18 - House of Quality

3 Implementação

“A mudança é fácil de propôr, difícil de implementar, e ainda mais difícil de suster”

Andy Hargreaves

Neste capítulo é realizada a análise e design da solução proposta, seguida de uma breve explicação do funcionamento da aplicação e dos seus requisitos. De seguida serão expostos os casos de uso do utilizador.

Será de seguida apresentado quais as soluções arquiteturais e tecnológicas pensadas e as razões que levaram à solução final.

A implementação e desenvolvimento da solução serão descritos, desde os primeiros protótipos no Unity até à utilização do Indoor Atlas e todo o processo de mapeamento do local.

3.1 Contexto da solução

Como referido anteriormente quando alguém chega a um determinado local pela primeira vez tende a sentir-se perdido e a precisar de ajuda para se orientar. Da mesma maneira quando se quiser deslocar para outro local, precisa de obter um trajeto, de preferência o mais rápido possível, para se deslocar do ponto A para o ponto B. Desta forma o objetivo da solução explorada nesta dissertação é ajudar o utilizador a deslocar-se desde um ponto A até a um Ponto B, como se evidencia pela figura 19.

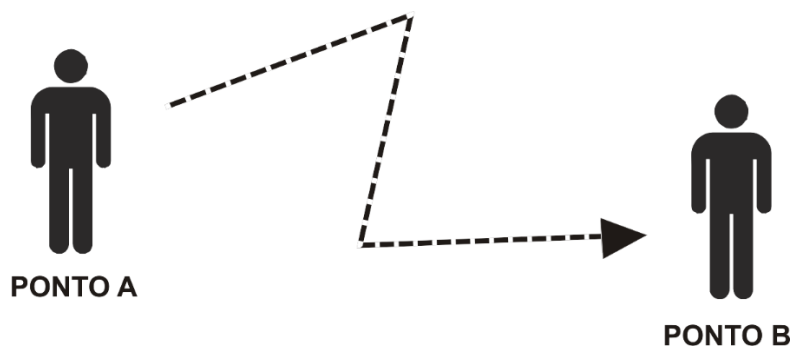


Figura 19 – Objetivo da aplicação

Caso o local onde o indivíduo se encontra seja relativamente simples, de maneira a que para chegar de A a B ele tenha apenas de seguir em frente numa direção, o caso é bastante simples pois basta-lhe mostrar uma rota ou dar-lhe uma explicação simples como “Seguir em frente até ao fim do edifício”. Mas caso o local em que a pessoa se encontra seja mais complexo, com por exemplo vários corredores ou cruzamentos, diversas salas no mesmo corredor ou até mesmo vários pisos, mostrar-lhe apenas uma rota não basta pois sendo que ele não conhece o ambiente em redor é provável que acabe por se enganar em determinada parte do percurso. Para combater isso é necessário que o utilizador seja periodicamente informado de qual o caminho a seguir. No caso desta solução, como está evidenciado na figura 20, estas direções são mostradas ao utilizador através de setas de orientação que irão aparecer, através de técnicas de Realidade Aumentada, posicionadas estrategicamente à frente dele, de maneira a que saiba sempre para onde terá de se dirigir.

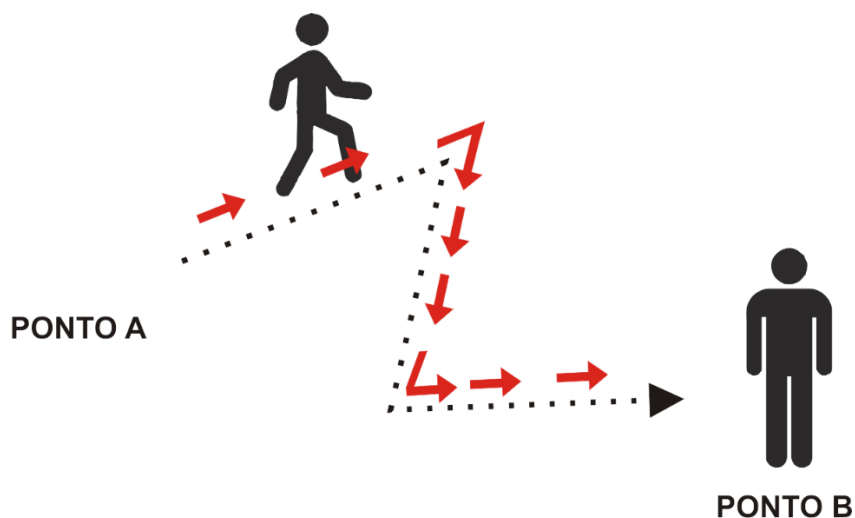


Figura 20 – Orientação do utilizador ao longo do percurso

É também necessário perceber se a deslocação será feita em ambiente aberto como num jardim ou numa cidade, dentro de um edifício ou de outro local fechado ou de pequenas dimensões, ou se uma mistura dos dois. Este conhecimento é necessário pois dependendo do tipo de deslocamento, existem diferentes técnicas de geolocalização e de orientação no espaço indicadas.

Como se pode perceber pelo esquema da figura 21, para a solução desenvolvida assume-se que toda a deslocação é realizada dentro do ISEP, mais concretamente dentro do edifício do Departamento de Engenharia Informática.

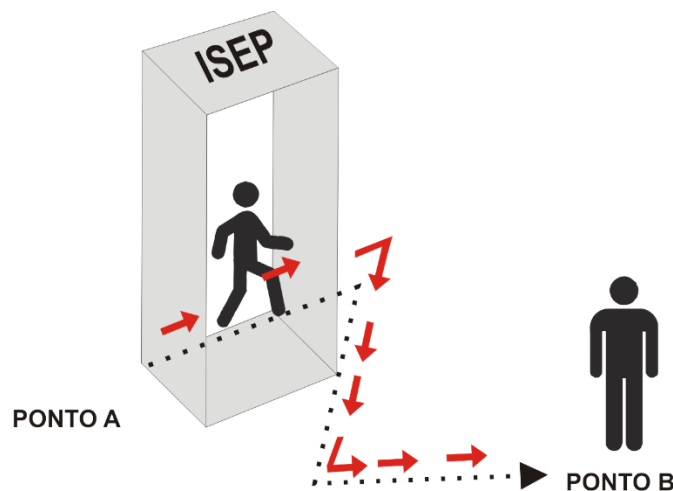


Figura 21 – Esquema que depreende que o percurso realizado acontece dentro do ISEP

3.2 Análise e Design da solução

Utilizando o levantamento dos requisitos feitos no capítulo da Análise de valor, foi feita a análise e design da solução proposta, dando origem aos casos de uso da aplicação e as várias ações que o utilizador pode realizar.

Foram também apresentadas as tecnologias utilizadas assim como as razões que levaram a mudar de tecnologias a meio do desenvolvimento, com o auxílio dos devidos diagramas de arquitetura de sistema.

3.2.1 Os Casos de Uso

Na Figura 22 encontra-se o diagrama de casos de uso. É possível visualizar que existem dois tipos de caso de uso. Os casos de uso de UC1 a UC4 são casos de uso considerados essenciais para o MVP e por isso prioritários. Os casos de uso de UC5 a UC8 são casos de uso extra a serem considerados após a criação do MVP.

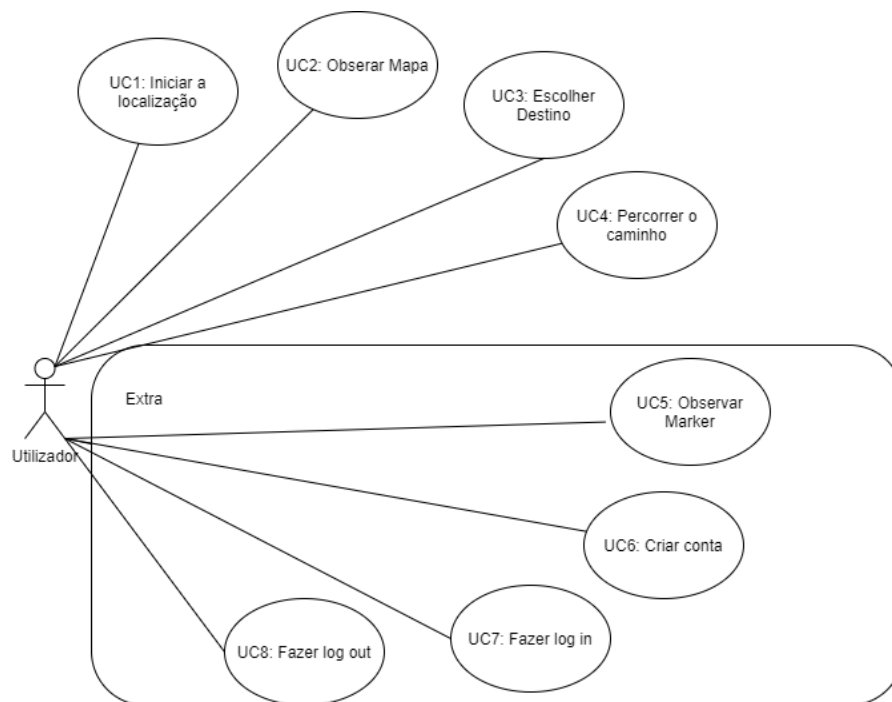


Figura 22 - Diagrama de casos de uso

3.2.1.1 UC1: Iniciar a localização

Neste caso de uso o utilizador irá permitir que a aplicação descubra a sua localização precisa, através do consentimento de uso do GPS.

3.2.1.2 UC2: Observar mapa

O utilizador pode observar um mapa que aponta para a local onde este se encontra e pode observar o mapa detalhado do andar do edifício, assim como o ponto do edifício em que ele se encontra, com um certo raio de incerteza.

3.2.1.3 UC3: Escolher o destino

Este caso de uso permite que o utilizador escolha um destino através de uma lista de locais. Ao clicar num elemento desta lista, poderá ver também a localização do destino no mapa.

3.2.1.4 UC4: Percorrer caminho.

O utilizador tem à sua disposição um indicador em Realidade Aumentada que mostra qual a direção a seguir. Pode também observar todo o caminho que tem de percorrer abrindo o mapa.

3.2.1.5 UC5: Observar marker

Neste caso de uso é dada a possibilidade de o utilizador usar a funcionalidade da camera com Realidade Aumentada para observar um marker que terá informação adicional sobre o local onde o utilizador se encontra atualmente.

3.2.1.6 UC6: Criar conta

Neste caso de uso o utilizador poderá criar uma conta para aceder à aplicação e ser reconhecido nesta.

3.2.1.7 UC7: Fazer log in

O utilizador poderá fazer log in na aplicação para ser identificado nesta e poder usufruir das suas funcionalidades.

3.2.1.8 UC8: Fazer log out

O utilizador tem a possibilidade de fazer log out na aplicação.

3.2.2 Diagramas de Arquitetura

A arquitetura do sistema define todos os componentes de software, como se relacionam entre si e com software externo. Para o nosso problema foram selecionadas duas soluções distintas. Na figura 23 está representada a arquitetura dos componentes

da primeira solução. Esta solução assenta em três componentes. A aplicação mobile, o servidor, e a Google API.

A aplicação mobile vai deter toda a logica de renderização e de geolocalização. Será responsável por toda a interação com o utilizador, assim como utilizar o pedometro e o giroscópio do dispositivo móvel para orientar o utilizador.

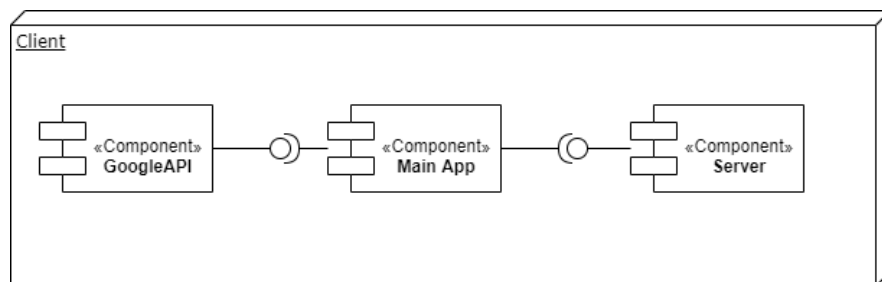


Figura 23 - Primeiro diagrama de arquitetura da solução

Esta aplicação móvel vai consumir o API da Google para mostrar o mapa ao utilizador e sobrepor outros mapas ou assinal a posição corrente do utilizador.

Por último a aplicação móvel poderá também consumir um servidor que estará responsável por toda a componente de registos e log ins dos utilizadores. Este componente será apenas utilizado caso se decida fazer os casos de uso UC6 a UC8.

A segunda solução, como podemos ver na figura 24, estende o primeiro diagrama, acrescentando-lhe a API da Indoor Atlas.

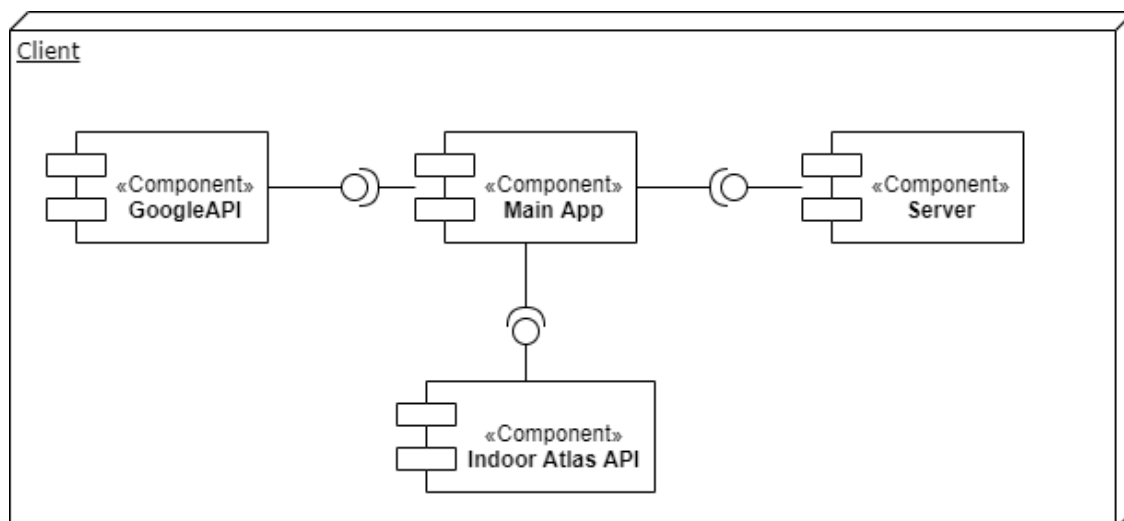


Figura 24 - Segundo diagrama de arquitetura da solução

Desta forma a responsabilidade da geolocalização deixava de fazer parte da aplicação mobile. Em vez disso esta passava a consumir a API da Indoor Atlas, fazendo pedidos de localização e recebendo respostas com a localização do utilizador e a rota a seguir para chegar ao seu destino.

Numa primeira fase foi utilizado a primeira solução visto que seria mais simples e requeria menos chamadas a API's externas, sendo por isso mais rápida a fornecer informação. Porém, como será explicado posteriormente, esta primeira solução não era fiável o suficiente e por isso teve de se optar pela segunda opção.

3.3 A Orientação no espaço

Como dito anteriormente há várias maneiras de localizar um utilizador no espaço e de o guiar. A primeira abordagem explorada pelo aluno foi a de guiar o utilizador utilizando o pedometro e giroscópio do dispositivo móvel para saber sempre em que posição o utilizador se encontra.

Este processo divide-se em três partes. A primeira parte consiste em criar um grafo com todos os pontos a que o utilizador se pode deslocar, com as devidas distâncias e caminhos entre os pontos. Este grafo caso deve ter dois tipos de nós. Nós principais que apontam para os diversos locais que o utilizador pode visitar e nós secundários ou auxiliares que são colocados sempre que há uma mudança de direção. Desta maneira é possível calcular o caminho mais rápido para ir do ponto A ao ponto B e é possível avisar o utilizador sempre que chegou a um nó secundário e portanto terá de mudar de direção.

A segunda é saber o ponto inicial em que o utilizador se encontra. Para isso duas alternativas foram estudadas. A primeira seria utilizar a própria localização do telemóvel para obter as coordenadas geográficas. Porém para além destas terem alguma imprecisão, era necessário uma maior complexidade para posicionar o utilizador dentro do grafo falado anteriormente. A segunda opção seria colocar um Markers em pontos específicos do DEI, que seriam também mapeados no grafo. O utilizador teria depois apontar a câmara para fornecer a sua localização inicial. Embora

esta segunda abordagem seja um pouco mais restritiva, isso seria facilmente resolvido aumentando o número de markers dentro do departamento.

Por último, sabendo a posição inicial do utilizador seria necessário estar a escutar o pedómetro e o giroscópio do dispositivo. Desta maneira seria possível detetar quando o utilizador estava em movimento e, através do giroscópio, qual a direção em que ele se estaria a deslocar, sendo assim possível calcular a distância até ao próximo nóculo do grafico. Porém, após serem realizados alguns testes, viu-se que esta abordagem não era viável pois além de o pedómetro ter um erro ainda elevado mesmo depois de calibrado, principalmente se o utilizador abrandasse ou acelerasse a passada habitual, o giroscópio apenas dá a direção para qual o dispositivo está a apontar, ou seja, o utilizador teria de estar com o telemóvel a direção na qual se está a dirigir ou a aplicação ia começar a dar dados errados.

Assim sendo uma outra abordagem teve de ser explorada.

3.3.1 A Utilização do Indoor Atlas

De forma a melhorar a precisão da aplicação, o uso do pedómetro e do giroscópio foi substituído pela integração com a API da indoor Atlas que permite obter as coordenadas geográficas do utilizador, com rigor, a qualquer altura.

Mas antes de ser possível realizar chamadas a esta API é necessário seguir vários passos. Antes de mais é necessário fazer o registo grátis no website da indoorAtlas como developer. Após esse passo, o utilizador da Indoor Atlas é levado para a sua página principal, como se pode ver na figura 25. Nesta página é possível gerir todas as aplicações do utilizador que utilizam a API, assim como mapear novas localizações ou gerir o posicionamento de beacons.

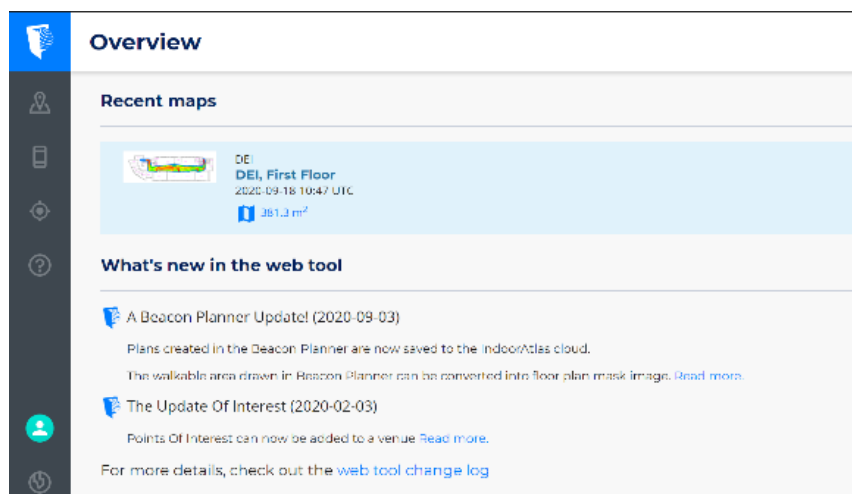


Figura 25 - Pagina principal do Indoor Atlas

Para o caso em específico desta aplicação é necessário criar, dentro do Indoor Atlas, a localização que pretendemos usar. Uma localização, para o Indoor Atlas, é um contentor para planos de pisos que estejam relacionados. Normalmente estas localizações correspondem a edificios e contem um ou mais planos para cada piso do edificio.

De forma a criar uma nova localização é necessário fornecer em primeiro lugar o nome da localização. Este nome não precisa necessariamente de ser o nome verdadeiro do local, é apenas um nome para o utilizador saber qual a localização que está a gerir. É possível também adicionar uma descrição do local para maior detalhe. É necessário também fornecer as coordenadas do edificio para que posteriormente o Indoor Atlas saiba se o utilizador está ou não efetivamente no perimetro da localização.

Após a identificação da localização e das suas coordenadas é necessário fornecer ao Indoor Atlas cada um dos planos do edificio. Para isso é necessário fornecer um nome para o plano, o número do andar, e uma altitude aproximada relativamente ao plano do chão. Por último é preciso fornecer uma imagem PNG ou JPEG do plano e alinha-la com o mapa fornecido pelo Indoor Atlas como se pode ver na figura 26. O plano pode ser escalado facilmente dentro do website e rodado de forma a que esteja o mais alinhado possível com o edificio, pois quanto melhor alinhado estiver o plano mais precisa vai ser depois a localização fornecida.

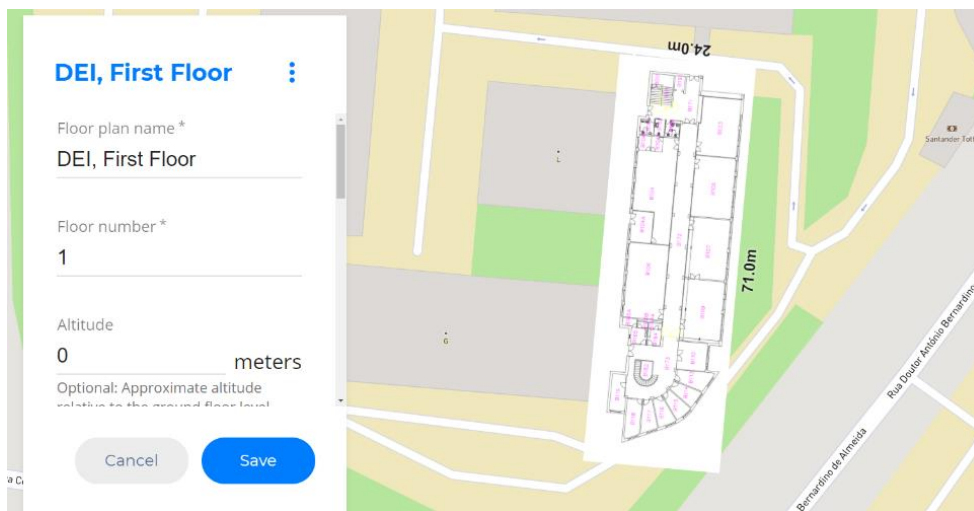


Figura 26 – Interface para alinhamento dos planos dos pisos no Indoor Atlas

Estando a localização criada, é necessário criar o grafo com todos os caminhos possíveis. Para isso é necessário em primeiro lugar entrar no modo de edição do edifício e entrar na aba de “Waypoints”. Nesta aba é necessário selecionar vários Waypoints para cada mapa. Um Waypoint é um qualquer lugar no espaço que vai ser utilizado para unir a localização física e do mapeador que vamos utilizar posteriormente e o nosso plano do piso. Estes pontos convem serem precisos e facilmente detetáveis na planta do piso, portanto alguns bons objetos para utilizar como Waypoints são cantos ou pilares do edifício, escadas, elevadores, portas entre outros. Na figura 27 é possível ver os waypoints utilizados para o piso do rés do chão do DEI.



Figura 27 – Mapa com os Waypoints criados

Estando os waypoints criados, é necessário ir para a aba de “Map data”. Neste aba inicialmente vai ser dito que não existem dados sobre o mapa e que é necessário utilizar a aplicação MapCreator 2.0 para gerar esses dados.

Depois de obtido o MapCreator é necessário ir ao edifício e começar a mapea-lo passando por cada um dos waypoints e percorrendo o caminho entre estes. No fim, entrando outra vez na aba de “Map data” é possível ver informações sobre os dados colecionados, como por exemplo a cobertura e qualidade do mapeamento magnético, a cobertura e qualidade do mapeamento por wi-fi e a qualidade do ambiente de beacons. No caso do DEI é possível ver na figura 28 a cobertura magnética no fim do mapeamento.

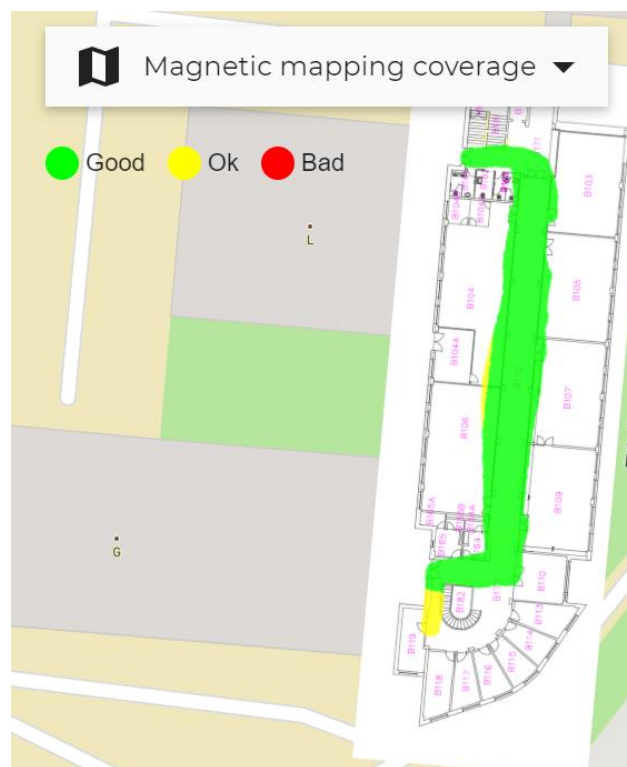


Figura 28 – Cobertura magnética do DEI

No fim de todos estes passos é possível clicar na aba de “Wayfinding” e testar vários pontos do mapa e observar qual o caminho que o Indoor Atlas recomenda. Caso não apareça nenhum caminho, e seja possível ir do ponto A ao ponto B, é necessário melhorar a qualidade dos waypoints e/ou do mapeamento de dados.

Por último, tendo todos os edifícios necessários e os seus pisos mapeados, é preciso ir à aba “My apps” e criar uma nova aplicação dando-lhe um novo e se necessário uma descrição. Esta aplicação serve apenas de um contentor para ser mais fácil de organizar as chaves da API por aplicação. Após a criação da aplicação é preciso pedir uma chave para a API. Estas chaves podem ser para utilização da API juntamente com o sdk da Indoor Atlas ou apenas para uso da data API que permite obter dados sobre o posicionamento do utilizador em cada sessão. No caso específico desta tese, a chave que melhor cumpre as funcionalidades necessárias é a chave para o SDK.

Obtendo assim todas as ferramentas necessárias para a orientação no espaço é necessário agora começar a criação da aplicação para dispositivos móveis e fazer a integração dessa aplicação com o SDK da Indoor Atlas.

3.4 Aplicação Mobile

Como dito no capítulo do estado da arte, numa primeira fase foi utilizado o motor de jogo Unity para começar a fazer a aplicação mobile.

Em primeiro lugar para trabalhar com Realidade Aumentada no Unity 2019 é necessário criar um novo projeto e inserir os plugins de Realidade Aumentada “AR Foundation” e “AR Core”. Para isso basta selecionar a aba “window”, “package manager” e instalar os plugins.

Após estas instalações começou a ser feita a *user interface* que inicialmente consistia numa combo box para selecionar os destinos, um botão para inicializar a viagem e um botão para entrar na captura de Markers, caso necessário. Foram também feitos alguns testes sobre como posicionar objetos dentro do mundo real e deteção de planos.

Após se perceber que não se iria poder utilizar o pedómetro e giroscópio do dispositivo para a localização do utilizador, foi necessário integrar o plugin do Indoor Atlas, que consiste nas classes e pastas visíveis na figura 29.



Figura 29 – Plugin do Indoor Atlas para Unity

Porém, após alguma investigação e testes, notou-se que embora fosse possível encontrar a posição do utilizador com através da classe Location, e se soubesse a orientação do utilizador, não era possível receber o mapa do edifício, o grafo com os locais e os caminhos, nem sequer as direções a seguir para chegar do lugar A ao lugar B. Após mais alguns testes e algumas conversas com a equipa da Indoor Atlas, descobriu-se que o plugin para o Unity não permitia utilizar as técnicas de wayfinding providenciadas pelo sdk da Indoor Atlas e que o plugin tinha sido descontinuado, portanto a Indoor Atlas não tinha intenções de estender as funcionalidades do plugin. Contudo confirmaram que todas essas funções estavam disponíveis no sdk que podia ser utilizado em aplicações nativas.

Com este bloqueio, e após algum tempo de consideração das outras possibilidades já estudadas, foi decidido tentar passar toda a aplicação mobile para uma aplicação android nativa, utilizando o Android Studio.

3.4.1 A Utilização do Android Studio

Como o aluno nunca tinha utilizado o Android Studio antes, foi necessário realizar alguns tutoriais primeiro para se habituar ao desenvolvimento nativo de android, utilizando a linguagem de programação Java visto que o aluno já estava habituado a esta linguagem por causa das cadeiras efetuadas durante a licenciatura.

De seguida foi explorado o demo fornecido pela Indoor Atlas sobre a integração do sdk com o Android Studio. Depois de tudo entendido um novo projeto foi criado com as configurações de gradle necessárias para o uso do sdk, como se pode ver na figura 30.

```

android {
    compileSdkVersion 28
    buildToolsVersion "29.0.2"
    // note: use gradle.properties in project root level to set your api credentials
    def apiKey = project.properties['indoorAtlasApiKey'] ?: "api-key-not-set"
    def apiSecret = project.properties['indoorAtlasApiSecret'] ?: "api-secret-not-set"
    def pubNubPublishKey = project.properties['pubNubPublishKey'] ?: "not-set"
    def pubNubSubscribeKey = project.properties['pubNubSubscribeKey'] ?: "not-set"
    def backgroundReportEndPoint = project.properties['backgroundReportEndPoint'] ?: ""
    defaultConfig {
        applicationId "com.indooratlas.android.sdk.examples"
        minSdkVersion 24
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        // avoid getting errors from malformed string resources if secret contains '%' chars
        resValue "string", "indooratlas_api_key", apiKey
        resValue "string", "indooratlas_api_secret", apiSecret.replaceAll("%", "\\u0025")
        resValue "string", "pubnub_publish_key", pubNubPublishKey
        resValue "string", "pubnub_subscribe_key", pubNubSubscribeKey
        resValue "string", "background_report_endpoint", backgroundReportEndPoint
    }
}

dependencies {
    implementation 'de.javagl:obj:0.2.1'
    implementation 'com.indooratlas.android:indooratlas-android-sdk:3.2.1@aar'
    implementation 'com.android.support:design:28.0.0'
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.google.android.gms:play-services-maps:8.1.0'
    implementation 'com.google.maps.android:android-maps-utils:0.3.+@aar'
    implementation 'com.squareup.picasso:picasso:2.5.2'
    implementation 'com.davemorrissey.labs:subsampling-scale-image-view:3.2.0'
    implementation 'com.pubnub:pubnub-android:3.7.5'
}

```

Figura 30 - Configuração de gradle necessária para utilizar a API Indoor Atlas

Foram também introduzidas no ficheiro gradle.properties a chave e o segredo da API, que foram obtidos através do website como referido em cima.

Após todas as configurações estarem feitas, foi criada uma actividade e um layout para essa actividade. Nessa actividade foi criado uma combo box para a seleção do destino, um “SupportMapFragment” que é um objeto que detêm o mapa do Google Maps, um botão para mostrar ou esconder o mapa e por último um botão para iniciar a rota.

A actividade criada vai ser responsável por controlar o fluxo da aplicação e inicializar todos os serviços necessários para interagir com a API do Indoor Atlas e pedir o mapa à API da Google.

De forma a obter o mapa global é necessário estar à escuta do evento onMapReady. É através deste evento que se obtém a referencia do mapa, assim como se configura todas as settings necessárias como por exemplo desabilitar a localização do utilizador, caso se queira tratar disso manualmente, esconder a barra de ferramentas, ou

adicionar comportamento quando o utilizador clica no mapa ou num dos marcadores que estão dentro do mapa.

Convém também, para questões de performance, criar um `IARegionListener` que vai estar à escuta de quando o utilizador entra ou sai de uma localização, o qual a Indoor Atlas define como região. Desta maneira é possível saber quando o utilizador está perto de um edifício mapeado e assim poder mostrar apenas a planta desse edifício quando o utilizador se aproxima deste. Para renderizar a planta do edifício no mapa é necessário ir buscar o url que está na classe `FloorPlan` enviado pela API da Indoor Atlas e carregar o bitmap guardado nesse url. Para fazer esse carregamento foi utilizado a biblioteca Picasso, seguindo os passos demonstrados na figura 31. Por último é preciso adicionar esse bitmap ao mapa global da aplicação, da maneira que está na figura 32.

```
private void fetchFloorPlanBitmap(final IAFloorPlan floorPlan) {
    if (floorPlan == null) {...}
    mMap.getUiSettings().setCompassEnabled(true);
    final String url = floorPlan.getUrl();
    mLoadTarget = new Target() {
        @Override
        public void onBitmapLoaded(Bitmap bitmap, Picasso.LoadedFrom from) {
            if (mOverlayFloorPlan != null && floorPlan.getId().equals(mOverlayFloorPlan.getId())) {
                setupGroundOverlay(floorPlan, bitmap);
            }
        }
        @Override
        public void onPrepareLoad(Drawable placeHolderDrawable) {}
        @Override
        public void onBitmapFailed(Drawable placeHolderDrawable) {...}
    };
    RequestCreator request = Picasso.with(this).load(url);
    final int bitmapWidth = floorPlan.getBitmapWidth();
    final int bitmapHeight = floorPlan.getBitmapHeight();
    if (bitmapHeight > MAX_DIMENSION) {...} else if (bitmapWidth > MAX_DIMENSION) {...}
    request.into(mLoadTarget);
}
```

Figura 31 – Código para carregar o bitmap

```
private void setupGroundOverlay(IAFloorPlan floorPlan, Bitmap bitmap) {
    if (mGroundOverlay != null) mGroundOverlay.remove();
    if (mMap != null) {
        BitmapDescriptor bitmapDescriptor = BitmapDescriptorFactory.fromBitmap(bitmap);
        IALatLng iaLatLng = floorPlan.getCenter();
        LatLng center = new LatLng(iaLatLng.latitude, iaLatLng.longitude);
        GroundOverlayOptions fpOverlay = new GroundOverlayOptions()
            .image(bitmapDescriptor)
            .zIndex(0.0f)
            .position(center, floorPlan.getWidthMeters(), floorPlan.getHeightMeters())
            .bearing(floorPlan.getBearing());
        mGroundOverlay = mMap.addGroundOverlay(fpOverlay);
    }
}
```

Figura 32– Código para inserir o bitmap no mapa

Para conseguir receber informações sobre a localização do utilizador é necessário criar um `ILocationListener` que vai escutar todos os eventos da de “locationChanged” vindos da Indoor Atlas e vai tratá-los. Neste caso sempre que se obtem uma atualização da localização precisamos de verificar se o utilizador ainda está no mesmo piso e atualizar a posição do utilizador no mapa, como pode ser visto na figura 33.

```
private ILocationListener mListener = new ILocationListenerSupport() {
    @Override
    public void onLocationChanged(ILocation location) {
        if (mMap == null) {...}
        final LatLng center = new LatLng(location.getLatitude(), location.getLongitude());
        final int newFloor = location.getFloorLevel();
        if (mFloor != newFloor) {
            updateRouteVisualization();
        }
        mFloor = newFloor;
        showLocationCircle(center, location.getAccuracy());
        canCreateObject = true;
    }
    @Override
    public void onStatusChanged(String provider, int status, Bundle extras) {...}
};
```

Figura 33– Código que trata da mudança de localização

É também necessário criar um `IwayFindingListener` que é o objeto responsável por reagir sempre que há um update na rota tomada, como por exemplo uma mudança de direção. Sempre que recebemos um evento deste tipo é necessário verificar se o

utilizador já chegou ao destino e em caso afirmativo remover este listener. Caso contrário é preciso guardar a nova direção e no fim dar update à rota marcada no mapa. Passando agora à escolha dos destinos, a combo box referida em cima é preenchida através de um ficheiro de texto que é lido ao arrancar a aplicação, visto que o sdk para a orientação no espaço não fornece uma lista com os Waypoints definidos no grafo, nem permite que se dê nomes acessíveis a esses waypoints. Este ficheiro está guardado nos assets da aplicação e é facilmente modificado, porém num futuro poderá ser modificado para outro tipo de ficheiro ou até mesmo uma base de dados, caso se decida criar vários locais que o utilizador pode visitar, como por exemplo todos os edificios do ISEP. Sempre que um destino é selecionado é possível ver no mapa um marcador com a posição onde ele se encontra. Para isso é necessário remover o antigo marcador, se este existir, e adicionar o novo tal como se pode ver na figura 34.

```
spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {  
    @Override  
    public void onItemSelected(AdapterView<?> parentView, View selectedItemView, int position, long id) {  
        String location = (String)spinner.getSelectedItem();  
        double[] coordinates = _locations.get(location);  
        LatLng end = new LatLng(coordinates[0],coordinates[1]);  
        if(mMap!=null) {  
            if(mDestinationMarker!=null)mDestinationMarker.remove();  
            mDestinationMarker = mMap.addMarker(new MarkerOptions().position(end)  
                .icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_BLUE)));  
        }  
    }  
    @Override  
    public void onNothingSelected(AdapterView<?> parentView) {}  
});
```

Figura 34– Código para mudar marcadores

Por último, quando tudo o necessário para a orientação do utilizador está carregado e o utilizador selecionou um destino, o botão de start fica visível. Ao clicar nesse botão é iniciado um pedido à API do Indoor Atlas para iniciar a rota e estar à escuta das alterações de rota referidas anteriormente.

Após a conclusão de todo este processo e de testar a cada passo se os eventos estavam a ser chamados e se as direções recebidas eram de facto os corretos através de testes manuais, falta apenas adicionar o componente de Realidade Aumentada.

3.5 A Implementação da Realidade Aumentada

Para se perceber a integração do google ARCore numa aplicação de android nativo, foi primeiro estudado o demo providenciado pela google que contém os básicos das configurações do gradle para incluir o ARCore, assim como exemplos básicos para a renderização de objetos no mundo real e de deteção de planos para colocar esses objetos.

Pegou-se depois na aplicação mobile e acrescentou-se as configurações necessárias para a instalação do ARCore na aplicação, o que resultou no ficheiro detalhado na figura 35.

```
apply plugin: 'com.android.application'
android {...}
dependencies {
    implementation 'com.google.ar:core:1.18.0'
    implementation 'de.javagl:obj:0.2.1'
    implementation "com.indooratlas.android:indooratlas-android-sdk:3.2.1@aar"
    implementation 'com.android.support:design:28.0.0'
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.google.android.gms:play-services-maps:8.1.0'
    implementation 'com.google.maps.android:android-maps-utils:0.3.+
    implementation 'com.squareup.picasso:picasso:2.5.2'
    implementation 'com.davemorrissey.labs:subsampling-scale-image-view:3.2.0'
    implementation 'com.pubnub:pubnub-android:3.7.5'
    implementation 'org.osmdroid:osmdroid-android:6.0.3'
    implementation 'com.github.MKergall:osmbonuspack:6.5.2'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    implementation 'com.android.support:gridlayout-v7:28.0.0'
```

Figura 35 - Configuração de gradle necessária para utilizar o ArCore

Após toda a configuração estar feita foi necessário inicializar os componentes necessários para o uso de Realidade Aumentada, como o TapHelper para simular batidas no ecrã, que irão posteriormente ser utilizadas para a deteção de planos, e a GLSurfaceView que é onde vamos efetivamente renderizar os nossos objetos.

De seguida foi criado o método onDrawFrame que é responsável por a cada frame desenhar tudo o que está na nossa cena. Neste método verifica-se se todas as condições necessárias para a renderização estão validadas e em caso positivo, renderiza-se tudo o que é necessário à cena, desde texturas do background, a malhas de triângulos que

representam os planos onde as setas irão ficar. É também neste método que se atualiza as matrizes de transformação dos Anchors onde cada seta está ancorada. Uma visão global deste método pode ser visto na figura 36.

```
session.setCameraTextureName(backgroundRenderer.getTextureId());
frame = session.update();
camera = frame.getCamera();
if (frame.hasDisplayGeometryChanged() || calculateUVTransform) {...}
if (canCreateObject){
    createOrientationObject(frame, camera);
    canCreateObject = false;
}
backgroundRenderer.draw(frame, depthSettings.depthColorVisualizationEnabled());
trackingStateHelper.updateKeepScreenOnFlag(camera.getTrackingState());
if (camera.getTrackingState() == TrackingState.PAUSED) return;
float[] projmtx = new float[16];
camera.getProjectionMatrix(projmtx, 0, 0.1f, 100.0f);
float[] viewmtx = new float[16];
camera.getViewMatrix(viewmtx, 0);
final float[] colorCorrectionRgba = new float[4];
frame.getLightEstimate().getColorCorrection(colorCorrectionRgba, 0);
try (PointCloud pointCloud = frame.acquirePointCloud()) {...}
planeRenderer.drawPlanes(
    session.getAllTrackables(Plane.class), camera.getDisplayOrientedPose(), projmtx);
float scaleFactor = 1f;
virtualObject.setUseDepthForOcclusion(context: this, depthSettings.useDepthForOcclusion());
for (ColoredAnchor coloredAnchor : anchors) {
    if (coloredAnchor.anchor.getTrackingState() != TrackingState.TRACKING) {
        continue;
    }
    coloredAnchor.anchor.getPose().toMatrix(anchorMatrix, 0);
    virtualObject.updateModelMatrix(anchorMatrix, scaleFactor);
    virtualObject.draw(viewmtx, projmtx, colorCorrectionRgba, coloredAnchor.color);
}
```

Figura 36 – Implementação do OnDrawFrame

De forma a orientar o utilizador pelo espaço, foi decidido utilizar setas de direção. Porém como a implementação base do ARCore utiliza ficheiros obj juntamente com ficheiro png para as texturas, foi complicado encontrar um modelo que pudesse ser utilizado. Por isso foi utilizado a ferramenta Blender para criar o modelo de uma seta 3D e depois extraído a textura como png. Futuramente poderá-se utilizar bibliotecas que estender o ARCore, para melhorar e facilitar este processo, assim como abstrair as chamadas diretas a OpenGL.

Por último é necessário sempre que é verificada uma alteração da direção da rota, criar um novo Anchor posicionado à frente do utilizador e na orientação que ele tem de seguir. Para isso simula-se um toque no centro do ecrã. Através desse toque consegue-se descobrir se esse toque interceitou algum dos planos encontrados no mundo real, estes planos podem ser mesas, paredes, chão, tecto, qualquer superfície plana desde que o ARCore detete como tal. Em caso afirmativo é então criado um objeto do tipo Pose que detêm a posição e orientação no espaço. A posição do Pose é então a posição da interceção do toque com o plano e a orientação é obtida transformando a direção fornecida previamente pela rota do Indoor Atlas num Quaternion. Por último é então adicionado um novo Anchor, com o Pose criado, à lista de anchors que será depois chamada no método onDrawFrame. Todo este algoritmo pode ser observado na figura 37.

```

if (anchors.size() >= 20) {...}
long downTime = SystemClock.uptimeMillis();
long eventTime = SystemClock.uptimeMillis() + 100;
int metaState = 0;
MotionEvent motionEvent = MotionEvent.obtain(downTime, eventTime, MotionEvent.ACTION_UP,
    x: surfaceView.getWidth()/2, y: surfaceView.getHeight()/2, metaState
);
surfaceView.dispatchTouchEvent(motionEvent);
float[] objColor = new float[]{139.0f, 195.0f, 74.0f, 255.0f};
MotionEvent tap = tapHelper.poll();
if (tap != null && camera.getTrackingState() == TrackingState.TRACKING) {
    for (HitResult hit : frame.hitTest(tap)) {
        // Check if any plane was hit, and if it was hit inside the plane polygon
        Trackable trackable = hit.getTrackable();
        if ((trackable instanceof Plane
            && ((Plane) trackable).isPoseInPolygon(hit.getHitPose())
            && (PlaneRenderer.calculateDistanceToPlane(hit.getHitPose(), camera.getPose()) > 0))
            || (trackable instanceof Point
            && ((Point) trackable).getOrientationMode()
            == Point.OrientationMode.ESTIMATED_SURFACE_NORMAL)) {
            Anchor anchor = hit.createAnchor();
            double theta = Math.toRadians(mNextDirection)/2;
            float[] rotation = {0f, (float) Math.sin(theta), 0f, (float) Math.cos(theta)};
            Pose rotationPose = Pose.makeRotation(rotation);
            Pose directionPose = anchor.getPose().extractTranslation().compose(rotationPose);
            anchors.add(new ColoredAnchor(session.createAnchor(directionPose), objColor));
            this.runOnUiThread(this::showOcclusionDialogIfNeeded);
            break;
        }
    }
}
}

```

Figura 37 – Código para criar as setas de orientação

Todos estes passos foram executados em paralelo com testes manuais, porém, devido às adaptações referidas na secção “Adaptações”, alguns problemas com a integração e fiabilidade das informações fornecidas pela API, não foram encontradas até uma semana antes do término da dissertação, altura em que foi possível realizar um teste dentro do ISEP.

Na figura 38 é possível ver o primeiro destes testes que consiste em abrir a aplicação, entrar dentro do mapa e verificar que a aplicação deteta corretamente a posição do utilizador. Esta localização é vista pelo marcador no mapa, e pelo círculo à volta deste marcador, que delimita a área de possível erro da localização. Como se consegue perceber por esta imagem, o raio de incerteza da aplicação é mais ao menos de 1.5 metros, sendo possível reduzir este valor se for introduzida na API do indoor Atlas os pontos de acesso de Wifi ou através da introdução de beacons.

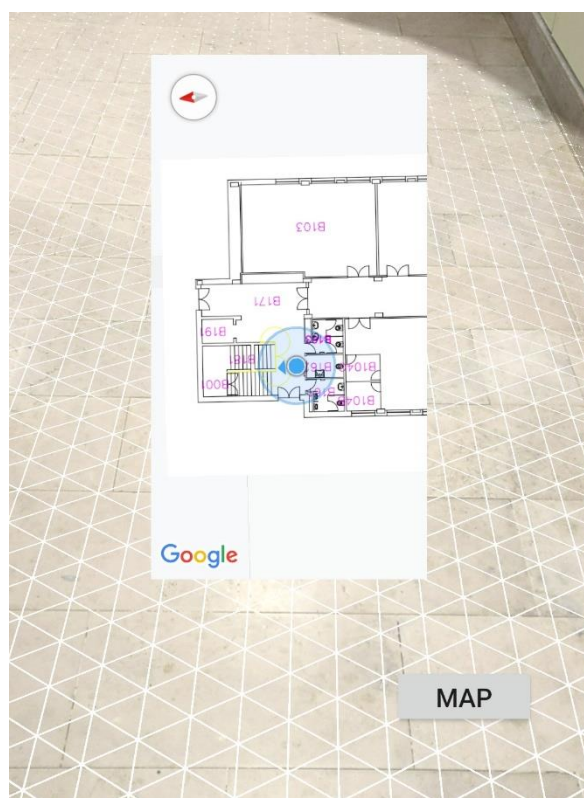


Figura 38 – Verificação da localização do utilizador

Depois, é necessário seleccionar um destino. Após essa interação um dos testes realizados foi abrir o mapa e verificar que é mostrado um percurso desde o local onde o utilizador se encontra até ao destino escolhido. Esta rota, como está ilustrado na

figura 39, é dado pelo marcado que representa o utilizador, um caminho traçado a azul escuro e por último um outro marcador que define o local de destino.

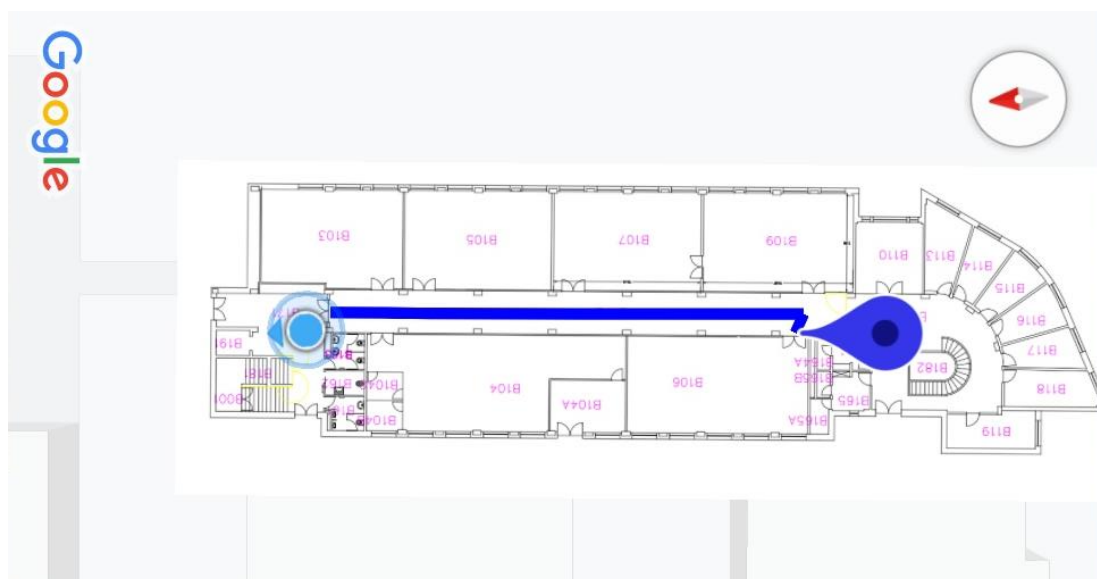


Figura 39 – Apresentação da rota no mapa

De seguida é iniciada a deslocação e verificado se a aplicação começa a colocar setas no chão apontando para a sala, como se pode observar na figura 40. Nesta figura pode-se também observar dois objetos na cena que são apenas auxiliares para testagem. O primeiro é uma malha de triângulos que demonstra o plano encontrado para colocar a seta. Esta malha é bastante útil para perceber se o utilizador vai conseguir observar a seta no chão ou não. O segundo é uma barra de mensagens que serve para observar certas informações em tempo real. No caso específico da figura 40 é possível observar a direção para o qual o utilizador está a apontar o telemóvel, assim como a direção que ele tem de seguir. Estas direções são dadas em graus sendo que o grau 0º representa o Norte.

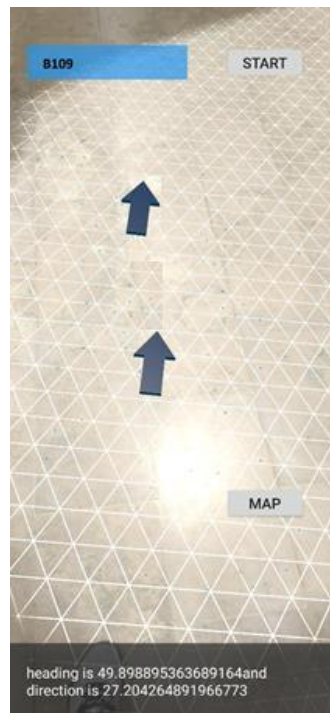


Figura 40 – Iniciação da deslocação

Por último é necessário percorrer o caminho todo e ver se as setas vão apontando na direção correta. Na figura 41 é possível ver um dos trajetos feitos. A quantidade de setas renderizadas deve-se ao facto de se estar a renderizar uma seta quando se deteta mudanças de localização e das já faladas incongruências de respostas da API.



Figura 41 – Conclusão da deslocação

3.6 Adaptações

Devido à situação pandémica global várias alterações aos âmbitos e objetivos do projeto tiveram de ser adaptados.

Em primeiro lugar, devido ao facto do local de desenvolvimento do projeto ter sido inicialmente no Laboratório de Multimédia(LAMU) do ISEP, que esteve fechado durante vários meses, teve de se adaptar todo o ambiente de desenvolvimento assim como as ferramentas utilizadas como dispositivos móveis e computadores, o que demorou várias semanas para que tudo estivesse tratado. Por causa uma adaptação aos objetivos teve de ser feita, ficando decidido por causa disso a não implementação dos casos de uso extra.

Também devido a este problema todos os testes realizados à aplicação tiveram de ser feitos em casa deste em vez de no ISEP como inicialmente pensado, tendo sido apenas possível realizar um teste final no ISEP e impedindo qualquer desenvolvimento de avaliações alpha e beta. De maneira a poder testar a aplicação, foi necessário arranjar uma planta da casa do aluno. Todo este processo foi também demorado pois a junta de freguesia não colaborou, obrigado o aluno a utilizar programas grátis que utilizam o dispositivo móvel para fazer scan às diferentes divisões do apartamento, o que por si é um processo lento e ainda mais complicado se o apartamento em si tiver mobilado de maneira a impedir de dar scan aos alicerces da casa.

3.7 Resumo

Neste capítulo fez-se, primeiramente, uma contextualização da solução assim como uma análise da aplicação e divisão das suas partes funcionais: Realidade Aumentada e orientação no espaço. Foram também estudados os casos de uso que representam a interação entre o utilizador e o sistema assim como as diferentes arquiteturas possíveis para a solução.

Em segundo lugar, abordou-se as diferentes técnicas de localização e orientação no espaço, e quais as decisões que levaram ao uso da API Indoor Atlas.

Posteriormente foi detalhado todo o desenvolvimento da aplicação mobile, a escolha de fazer uma aplicação android nativa, e a integração com o sdk da Indoor Atlas. Foram também mostrados trechos do código utilizado para fazer esta integração.

De seguida, foi abordado a integração da Realidade Aumentada e do sdk ARCore na aplicação mobile, desde a configuração inicial até aos métodos necessários para renderizar os objetos necessários na cena.

Por fim, foi também abordado os impactos do COVID-19 no processo de desenvolvimento de toda a aplicação.

4 A Avaliação da Solução

“O verdadeiro gênio reside na capacidade de avaliar a incerta, difícil e conflituosa informação ”

Winston Churchill

Este capítulo tem como objetivo detalhar como será realizada a avaliação da solução. Será apresentado o *Qualitative Evaluation Framework* (QEF). Será também referenciado as metodologias de avaliação pensados para a solução e os motivos para os quais não ter sido possível realiza-las.

4.1 QEF

O *Qualitative Evaluation Framework* (QEF) é uma ferramenta usada na avaliação de um produto e permite medir quantitativamente a qualidade de um sistema.

É composta por um plano multidimensional, normalmente tridimensional, para representar o progresso das dimensões. No caso deste projeto, serão avaliadas a Funcionalidade, Adaptabilidade e Eficiência. O resultado pode ser consultado na tabela 42.

Cada dimensão é dividida em fatores. Estes fatores estão divididos em diversos requisitos.

Cada dimensão têm um peso específico, de forma a assinalar a importância real de cada requisito. Cada requisito tem associadas métricas de avaliação rigorosas que permitem medir com precisão se o requisito cumpre ou não o estabelecido.

q	D	qi	Dimensão	Qi	W _{ij} (Peso do Fator j em Dim i) [0,1]	Fator	rw _{jk} (peso do requisito k em Fator j) {2, 4, 6, 8, 10}	Requisito	wf _k % nível conclusão do requisito k) [0,100]
0%	1,73	0	Funcionalidade	0	0,50	Funcional (Referentes a Casos de Uso)	10	FF01 - Indicar o ponto de partida	0
							10	FF02 - Escolher o destino	0
							10	FF03 - Percorrer o caminho	0
							10	FF04 - Observar Marker	0
							10	FF05 - Fazer uma chamada	0
							5	FF06 - Deixar mensagens	0
							5	FF07 - Criar conta	0
							5	FF08 - Fazer log in	0
							5	FF09 - Fazer log out	0
				0	0,22	Interação com o Utilizador	10	FI01 - A aplicação é intuitiva	0
							8	FI02 - É possível ver uma lista dos vários locais	0
							10	FI03 - Uma seta direcional indica ao utilizador a direção a seguir	0
							8	FI04 - É possível ver a distância a que se está de cada local	0
				0	0,28	Qualidade do Conteúdo	8	FQ01 - Toda a informação do produto está bem organizada	0
							8	FQ02 - Os textos estão bem escritos e as frases fazem perfeito sentido	0
							10	FQ03 - Todos os widgets estão acessíveis e não sobrepostos	0
							10	FQ04 - A seta direcional aponta para a direção a seguir	0
							8	FQ05 - Todos os conteúdos são relacionados com o produto	0
		0	Adaptabilidade	0	0,50	Versatilidade	10	AV01 - A aplicação é apresentada de forma nativa ao ambiente operacional	0
				0	0,25	Aspeto Pedagógico	10	AV02 - A dimensão da aplicação adequa-se ao tamanho do dispositivo	0
				0	0,25	Manutenção	8	AA01 - A aplicação fornece uma informação extra sobre os locais visitados	0
		0	Eficiência	0	0,20	Qualidade visual	8	AM01 - A aplicação permite a adição de markers com informação nova	0
							10	EQ01 - Os elementos virtuais interligam-se bem com os elementos reais	0
				0	0,80	Navegação	10	EN01 - A aplicação calcula a distância entre locais de forma rápida e eficiente	0
							10	EN02 - A aplicação calcula o caminho mais curto de forma rápida e eficiente	0
							8	EN02 - A interface da aplicação é rápida e responsiva	0
							8	EN03 - A aplicação não tem erros ao correr e erros inesperados devem também ser tratados	0

Figura 42 - Qualitative Evaluation Framework

4.2 A Metodologia de Avaliação

A metodologia usada para desenvolver e avaliar a solução será recorrendo a um MVP. Este MVP terá um conjunto de funcionalidades fulcrais para o funcionamento da aplicação (UC1 – UC5). Após este MVP ser concluído realizar-se-á o primeiro elemento de avaliação. Seguidamente ir-se-á atuar sobre o feedback desse momento de avaliação e continuar a desenvolver os casos de uso extra (UC6-UC9) até a aplicação estar na sua etapa de desenvolvimento final e se fazer um segundo momento de avaliação.

A solução será avaliada através recorrendo ao QEF e às tabelas 4 a 9, que se encontram a seguir. Estas tabelas definem o nível de cumprimento de cada tarefa num valor de 0 a 100 dependendo da métrica de avaliação definida para cada requisito.

Para o requisito FI01 um questionário composto por várias questões sobre a interface gráfica da aplicação será criado e preenchido por utilizadores escolhidos como “beta testers”. A avaliação deste requisito será dada pela avaliação global destes utilizadores.

Tabela 4 - Métricas de avaliação da Funcionalidade

Requirement	Metric Evaluation	Wfk - Fullfilment (%)		
		0	50	100
FF01 - Indicar o ponto de partida	O utilizador liga o GPS/ clica no local onde se encontra	Não implementado		Implementado
FF02 - Escolher o destino	O utilizador tem uma lista de destinos para escolher	Não implementado		Implementado
FF03 - Percorrer o caminho	O utilizador é orientado até ao destino	Não implementado	Algumas falhas de orientação	Implementado
FF04 - Observar Marker	O utilizador olha para um marker e informação extra aparece	Não implementado		Implementado
FF05 - Fazer uma chamada	O utilizador clica num número e é redirecionado para a aplicação de chamadas do seu sistema operativo	Não implementado		Implementado
FF06 - Deixar mensagens	O utilizador consegue ler e escrever mensagens no mundo	Não implementado	Apenas mensagens que ele escreveu	Implementado
FF07 - Criar conta	O utilizador consegue criar uma conta	Não implementado		Implementado
FF08 - Fazer log in	O utilizador consegue fazer log in	Não implementado		Implementado
FF09 - Fazer log out	O utilizador consegue fazer log out	Não implementado		Implementado

Tabela 5 - Métricas de avaliação da interação com o utilizador

Requirement	Metric Evaluation	Wfk - Fullfilment (%)		
		0	50	100
FI01 - A aplicação é intuitiva	Questionário de interação a 2 tipos de utilizador. O primeiro que leu o manual e o segundo que não o leu. São dados questionários a 3 elementos de cada tipo num total de 6.	0-1 questionário positivo	2-3 questionários positivos	4-6 questionários positivos
FI02 - É possível ver uma lista dos vários locais	A lista de todos os locais é visível	Lista existente	Alguns locais não estão registados	Todos os locais aparecem
FI03 - Uma seta direcional indica ao utilizador a direção a seguir	A seta indica a direção que o utilizador deve tomar	Não existe seta	Existem alguns erros na direção da seta	A seta aponta corretamente
FI04 - É possível ver a distância a que se está de cada local	Distância	Não é possível ver a distância	Existe alguma discrepância entre a distância real e a mostrada	É mostrada a distância correta

Tabela 6 - Métricas de avaliação da qualidade do conteúdo

Requirement	Metric Evaluation	Wfk - Fullfilment (%)		
		0	50	100
FQ01 - Toda a informação do produto está bem organizada	Toda a informação do produto deve ser dividida pelas áreas funcionais	Não	-	Sim
FQ02 - Os textos estão bem escritos e as frases fazem perfeito sentido	As frases são curtas e gramaticamente corretas	Não	-	Sim
FQ03 - Todos os widgets estão acessíveis e não sobrepostos	Os widgets estão visíveis e não colidem	Não	-	Sim
FQ04 - A seta direcional aponta para a direção a seguir	A seta aponta para a direção correta	Não	-	Sim
FQ05 - Todos os conteúdos são relacionados com o produto	Todo o conteúdo deve ser relacionado com a aplicação	Não	-	Sim

Tabela 7 - Métricas de avaliação da Adaptabilidade

Requirement	Metric Evaluation	Wfk - Fullfilment (%)		
		0	50	100
AV01 - A aplicação é apresentada de forma nativa ao ambiente operacional	Aplicação de Realidade Aumentada é nativa	Não	-	Sim
AV02 - A dimensão da aplicação adequa-se ao tamanho do dispositivo	A aplicação faz stretch ao ecrã	Não	-	Sim
AA01 - A aplicação fornece uma informação extra sobre os locais visitados	É possível obter informação sobre os locais visitados através de markers	Não	-	Sim
AM01 - A aplicação permite a adição de markers com informação nova	É possível adicionar novos markers	Não	-	Sim

Tabela 8 – Métricas de avaliação da Eficiência

Requirement	Metric Evaluation	Wfk - Fullfilment (%)		
		0	50	100
EQ01 - Os elementos virtuais interligam-se bem com os elementos reais	Os elementos virtuais estão bem visíveis e não destoam com o mundo real	Não	-	Sim
EN01 - A aplicação calcula a distância entre locais de forma rápida e eficiente	A aplicação deve mostrar de forma imediata a distância a que o utilizador se encontra dos locais	Não	-	Sim
E02- A aplicação calcula o caminho mais curto de forma rápida e eficiente	A aplicação deve mostrar de forma imediata o caminho mais curto	Não	-	Sim
EN02 - A interface da aplicação é rápida e responsiva	Funcionalidades que demorem mais do que 2 segundos devem mostrar um ícone que indique o seu progresso.	Não	-	Sim
EN03 - A aplicação não tem erros ao correr e erros inesperados devem também ser tratados	O utilizador deve receber mensagens de erro claras quando acontece um erro inesperado.	Não	-	Sim

Infelizmente, como referido anteriormente, todo este processo de avaliação e de testes alpha e beta não pode ser realizado devido à situação pandémica atual.

5 Conclusões e trabalho futuro

“A melhor maneira de lidar com o futuro é criando-o”

Ilya Prigogine

O último capítulo da presente dissertação comporta o balanço do projeto desenvolvido e implementado ao longo do ano letivo 2019/2020. Aqui são expostos os aspetos positivos assim como as dificuldades encontradas ao longo de todo o percurso. São também expostas possíveis melhorias e implementações do projeto.

5.1 Conclusões

O principal objetivo desta dissertação, que consistia na demonstração da ideia do uso da Realidade Aumentada aplicada à orientação em espaços fechados, foi atingido. No decorrer da investigação foi desenvolvida uma aplicação com recurso à Realidade Aumentada e geo-localização para guiar utilizadores dentro do DEI ISEP.

Efetou-se, primeiramente, um estudo da Realidade Aumentada, desde a sua história aos actuais usos e métodos de desenvolvimento.

Em segundo lugar realizou-se também um estudo sobre a orientação no espaço, assim como as diferentes técnicas de orientação e localização, quais os seus benefícios e quais os seus pontos fracos.

Durante o desenvolvimento da aplicação, foi necessário proceder a inúmeros testes para perceber se as metodologias indicadas eram precisas o suficiente para obter resultados satisfatórios ou se uma nova abordagem teria de ser encontrada.

Encontraram-se alguns obstáculos relativamente à integração de tecnologias, sendo que alguns dos recursos de Realidade Aumentada não suportavam as tecnologias de orientação no espaço necessárias.

Em suma, pode concluir-se que o objetivo inicial foi alcançado com sucesso pois foi demonstrado que é possível utilizar a Realidade Aumentada para orientar os utilizadores, desde que exista um fornecimento rigoroso de dados através de técnicas de orientação no espaço, existindo satisfação manifestada por parte dos intervenientes do Projeto. Desta forma, pode fazer-se uma autoavaliação positiva ao trabalho desenvolvido.

5.2 Trabalho futuro

Como trabalho futuro tem-se todos os casos de uso extra que não foram implementados, desde o scan de Markers e QR Codes ao serviço de troca de mensagens e de log-in. Pode também ser implementado um serviço de log-in de visitante para utilizadores que vão ao ISEP apenas uma vez, caso necessário. Para além disso pode ser ainda melhorado a estética da aplicação, assim como a maneira como está a ser implementado a Realidade Aumentada pois com o auxílio de algumas bibliotecas dedicadas é possível melhorar ainda mais a experiência do utilizador. É necessário também verificar a fiabilidade da Indoor Atlas API, e perceber se será necessário o suporte de mais algum tipo de técnica para conseguir orientar o utilizador com maior rigor.

A aplicação, como trabalho futuro, pode também ser aplicado a outros lugares como por exemplo os restantes edifícios do ISEP ou até mesmo alargar a aplicação a todo o Politécnico do Porto, sendo necessário para isso o mapeamento prévio de cada um dos lugares.

6 Referências

- ARToolKit. (2007). Obtido de <http://www.hitl.washington.edu/artoolkit/>
- ARToolKit. (2007). *ARToolKit - History*. Obtido de <https://artoolkit.org/about-artoolkit>
- Atlas, I. (s.d.). Obtido de Indoor Atlas: <https://www.indooratlas.com/>
- Awards, T. G. (2017). Obtido de The Game Awards: <https://thegameawards.com/awards>
- Brasil, F. (s.d.). *Ferramentas para otimizar avaliações internas*. Obtido de Medium: <https://medium.com/focobrasil/3-ferramentas-para-otimizar-avaliacoes-internas-932e4ae49a0b>
- Clemes, S., & Biddle, S. (s.d.). The Use of Pedometers for Monitoring Physical Activity in Children and Adolescents: Measurement Considerations. *Journal of Physical activity and health*.
- Developer, H. (s.d.). *Documentation*. Obtido de <https://developer.here.com/documentation>
- Developers, A. (s.d.). *What's new*. Obtido de Android Studio: <https://developer.android.com/studio/features>
- Developers, G. (s.d.). *ARCore*. Obtido de <https://developers.google.com/ar>
- Developers, G. (s.d.). *Overview*. Obtido de Geolocation API: <https://developers.google.com/maps/documentation/geolocation/overview>
- Faria, R., & Dias, E. (2007). Um estudo para a produção de conteúdos tridimensionais interativos.
- Games, E. (13 de December de 2019). *Senua's Saga: Hellblade II highlights a long list of Unreal Engine titles at The Game Awards 2019*. Obtido de Unreal Engine: <https://www.unrealengine.com/en-US/blog/senua-s-saga-hellblade-ii-highlights-long-list-of-unreal-engine-titles-at-the-game-awards-2019>
- Games, E. (s.d.). *Awards*. Obtido de Unreal Engine: <https://www.unrealengine.com/en-US/awards>
- Games, E. (s.d.). *Unreal*. Obtido de <https://www.unrealengine.com/en-US/>

Games, E. (s.d.). *Unreal® Engine End User License Agreement For Publishing*. Obtido de Unreal Engine: <https://www.unrealengine.com/en-US/eula/publishing?lang=en-US>

Geiger, C., Kleinnjohann, B., & Stichling, D. (2001). *Mobile AR4ALL*. Obtido de IEEE Explore: <https://doi.org/10.1109/ISAR.2001.970532>

Gerber, A., & Craig, C. (s.d.). *Learn Android Studio - Build Android Apps Quickly and Effectively*. Apress.

Google. (s.d.). *Google Glass*. Obtido de Google Glass: <https://www.google.com/glass/start/>

Google. (s.d.). *Street View*. Obtido de Google Maps: <https://maps.google.com/intl/pt-PT/streetview/>

Gregory, J. (s.d.). *Game Engine Architecture*. A K Peters, Ltd.

Group, W. M. (s.d.). *Quality Function Deployment*. Coventry, United Kingdom: University of WarWick.

Henrysson, A., Billinghurst, M., & Ollila, M. (s.d.). *Augmented Reality on Mobile Phones - Experiments and Applications*. Obtido de <https://ep.liu.se/ecp/016/007/ecp01607.pdf>

inspiration-log-3-the-ultimate-display. (15 de outubro de 2012). Obtido de sensoriumresearch: <https://sensoriumresearch.wordpress.com/2012/10/15/inspiration-log-3-the-ultimate-display/>

Institution, S. (13 de Januray de 2015). *Smithsonian Brings Historic Specimens to Life in Free "Skin and Bones" Mobile App*. Obtido de Smithsonian: <https://www.si.edu/newsdesk/releases/smithsonian-brings-historic-specimens-life-free-skin-and-bones-mobile-app>

Ionescu, D. (s.d.). *Geolocation 101: How It Works, the Apps, and Your Privacy*. Obtido de PCWorld: <https://www.pcworld.com/article/192803/geolo.html>

Koen. (2002). *Value Analysis: Meaning, Phases, Merits and Limitations*. New York, Wiley.

Library, Y. A. (18 de April de 2015). *Value Analysis: Meaning, Phases, Merits and Limitations*.

Martikainen, A. (2017). Front End of Innovation in Industrial Organization.

MAXST. (s.d.). Obtido de <http://maxst.com/#/en>

Microsoft. (s.d.). Obtido de <https://www.microsoft.com/en-us/hololens>

Microsoft. (s.d.). *Soluções prontas para negócios para o HoloLens 2*. Obtido de hololens: <https://www.microsoft.com/pt-br/hololens/apps>

Niantic. (s.d.). *Pokemon Go*. Obtido de <https://www.pokemongo.com/pt-pt/>

NOS. (s.d.). *Business Model Canvas*. Obtido de <https://www.nos.pt/empresas/repositorio-informacao/criar-uma-empresa/guias-teoricos/Pages/business-model-canvas.aspx>

Passaro, V., Cuccovillo, A., & Vaiani, L. (2017). Gyroscope Technology and Applications. A Review in the Industrial Perspective.

Pereira, D. (8 de June de 2016). *O que é o Business Model*. Obtido de O analista de modelos de negócios: <https://analistamodelosdenegocios.com.br/o-que-e-o-business-model-canvas/>

Protalinski, E. (8 de December de 2014). *Google releases Android Studio 1.0, the first stable version of its IDE*. Obtido de VentureBeat: <https://venturebeat.com/2014/12/08/google-releases-android-studio-1-0-the-first-stable-version-of-its-ide/>

Rodrigo Ramos, N. (2014). GOOGLE MAPS: UMA FERRAMENTA DE APOIO AO. XXI Semana de Geografia.

Santos, F., Feitosa, V., & Perinotto, A. (s.d.). Aplicativos de Mapas Google Maps, Here Maps e turismo.

Silva, D., Sales, G., & Castro, J. (2018). A Utilização do aplicativo plickers como ferramenta na implementação da metodologia peer instruction. *Revista Eletrônica Científica Ensino Interdisciplinar*.

Singapore, N. M. (18 de February de 2020). *Story of the Forest*. Obtido de nhb: <https://www.nhb.gov.sg/nationalmuseum/our-exhibitions/exhibition-list/story-of-the-forest>

SutherLand, I. (8 de August de 2005). *The ultimate display: where will all the pixels come from?* Obtido de ieeexplore: <https://ieeexplore.ieee.org/document/1492267>

The history of Google Glass. (s.d.). Obtido de <http://glassalmanac.com/history-google-glass/>

Unity. (2017). *Products*. Obtido de Unity: <https://unity3d.com/pt/unity>

Unity. (s.d.). *Store*. Obtido de Unity: <https://store.unity.com/>

Vroegop, D. (s.d.). Obtido de https://pt.linkedin.com/learning/app-development-for-microsoft-hololens?trk=seo_pp_d_cymbii_title_m015_learning

Vuforia Engine. (s.d.). Obtido de <https://developer.vuforia.com/>

Warren, T. (s.d.). *The Verge.* Obtido de <https://www.theverge.com/2019/8/8/20776247/google-maps-live-view-ar-walking-directions-ios-android-feature>

Wikitude. (s.d.). *Wikitude - About*. Obtido de <https://www.wikitude.com/about/>

Wu, H.-K., Lee, S. W.-Y., Chang, H.-Y., & Liang, J.-C. (March de 2013). Current status, opportunities and challenges of augmented reality in education.